Analysis of Clickstream Data

Amin Jamalzadeh

A Thesis presented for the degree of
Doctor of Philosophy

Statistics Group
Department of Mathematical Sciences
University of Durham
England

October 2011

Dedicated to

Professor Mohammad Salehi M.

Analysis of Clickstream Data

Amin Jamalzadeh

Submitted for the degree of Doctor of Philosophy
October 2011

Abstract

This thesis is concerned with providing further statistical development in the area of web usage analysis to explore web browsing behaviour patterns. We received two data sources: web log files and operational data files for the websites, which contained information on online purchases. There are many research question regarding web browsing behaviour. Specifically, we focused on the depth-of-visit metric and implemented an exploratory analysis of this feature using clickstream data. Due to the large volume of data available in this context, we chose to present effect size measures along with all statistical analysis of data. We introduced two new robust measures of effect size for two-sample comparison studies for Non-normal situations, specifically wherethe difference of two populations is due to the shape parameter. The proposed effect sizes perform adequately for non-normal data, as well as when two distributions differ from shape parameters. We will focus on conversion analysis, to investigate the causal relationship between the general clickstream information and online purchasing using a logistic regression approach. The aim is to find a classifier by assigning the probability of the event of online shopping in an e-commerce website. We also develop the application of a mixture of hidden Markov models (MixHMM) to model web browsing behaviour using sequences of web pages viewed by users of an e-commerce website. The mixture of hidden Markov model will be performed in the Bayesian context using Gibbs sampling. We address the slow mixing problem of using Gibbs sampling in high dimensional models, and use the over-relaxed Gibbs sampling, as well as forward-backward EM algorithm to obtain an adequate sample of the posterior distributions of the parameters. The MixHMM provides an advantage of clustering users based on their browsing behaviour, and also gives an automatic classification of web pages based on the probability of observing web page by visitors in the website.

Declaration

The work in this thesis is based on research carried out at the Statistics Group, the Department of Mathematical Sciences, Durham University, England. No part of this thesis has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Dr. David Wooff, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I am also thankful to my second supervisor, Dr. Peter Craig, who gave me useful guidance in several stages throughout my thesis with his patience and knowledge. This thesis would not have been possible without data set provided by SAYU company. I would also like to thank Gerda Arts who raised up the initial research question for me in this subject.

I gratefully acknowledge Mathematical science department for providing support and equipment I have needed to produce and complete my thesis. I would like to show my gratitude to Sharry Borgan for providing me the opportunity of gaining teaching experience during my study, as well as his support during teaching period. I have received a lot of support from IT specialists in the mathematical science department: Brian Rogers, Bernard Piette, and Mark Short, for which I am much appreciative. I would also like to thank our cleaning staff, Pamela and Michael, whose friendly smiles and morning hellos never failed to work our day.

I would like to thank my parents for all their love and encouragement. It was they who raised me with a love of science and supported me in all my pursuits and put me in the road of education. I offer special thanks to very my close friend, Shiler, who encouraged me to apply to Durham University and has been a great support during four years of living and studying at Durham. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this project.

Amin Jamalzadeh

Contents

List of Figures

List of Tables

Chapter 1

Clickstreams: New Source of Data in E-Commerce

1.1 Introduction

The adventure and advancement of the world wide web, beside all its influence on modern life, has played a major role to conduct business. Electronic commerce has grown rapidly in the decade by means of this new technology. Nowadays, firms offer goods/services not only through traditional channels such as retail outlets, but also in online virtual stores. An economic study (U.S. Census Bureau, 2005) conducted by the Department of Commerce shows that e-commerce, on a percentage basis, outperformed all four major economic sectors of manufacturing, merchant wholesalers, service industries, and retail trade in 2002-2003 (Banks and Said, 2006). However, e-commerce is more distinguished by changing the possibilities with regards to the distribution of goods or services. It has also served companies and organizations to improve their performance through better customer management, marketing strategies, and expanding the range of products and operations in the business area.

As the internet essentially works on the basis of data interchange, there is new data sources available which companies can exploit. This data enables e-commerce managers to supervise a business in ways that were not previously possible. Visitors' behaviour can be tracked by data collected in the server log files while they are surfing their website (Van den Poel and Buckinx, 2005). The prominent example of using this data, is so-called *Web Usage Mining*, which provides knowledge on how people behave in the web site specially in making purchase decisions. E-bay is an example in online auctions, which

provides data about the price of products by means of the amount of money customers are willing to pay. Another well-structured data set is produced by social network analysis, which helps to find relationships among people and business (Banks and Said, 2006). Bapna et al. (2006) illustrate automated data collection methods in several areas of e-commerce that give the resources for some tests of economic theories.

One of the favourable aspects of e-commerce is its ability to produce valuable data regarding website design, website performance and website customization. In investigating what needs to be fixed in an e-commerce website, customers play the most important role. Customers expect to be catered to, and they need a site that lets them get in and be immediately successful in pursuing their objectives. The e-retailer can use server logs data, instead of traditional customer surveys, to infer about customers' opinion regarding the website. For example, website managers by running A/B design tests are able to produce data about visitors' behaviour in two different designs of an webpage. A/B testing, or split testing, is a method by which a baseline control sample is compared to a variety of single-variable test samples in order to improve response rates. In the web designing context A/B split testing is exploited to determine which elements on a page are helping the performance of a web page, and which are not. For example, one might test two different headlines on a landing page and check whether one would outperform the other. The customers' usage information can help by providing a list of popular destinations from a particular webpage. The web manager is interested in investigating long convoluted traversal paths or low usage of a page with important site information by web usage data. The task may imply that the site links and information are not laid out in an intuitive manner (Cooley et al., 1999a).

The analysis of data generated by e-commerce provides the opportunity to outline better client relations (Bauer et al., 2002). It may be implemented more efficiently by influencing the current customers' and clients' visiting and shopping behaviour if the products and services get adjusted to the profile of visitors individually (Van den Poel and Buckinx, 2005). The more refined the segmentation or profiling of the customer base, the more efficiently a profitable target segment can be identified (Moe and Fader, 2004). By direct communication to current clients and prospective customers through websites, companies are able to adjust products, services, advertisement campaigns, and any other policies to the profile of visitors in order to influence customers' visiting and shopping behaviour. Baesens et al. (2004) outlined customer relationship management by considering alternative strategies pursued for different user segments, resulting from clustering methods on web data. Web data has been exploited to run systems which give advice about products, information or services a user might be interested in, while surfing the web pages.

These applications aim to assist users in a decision-making process where they want to choose some items amongst a potentially large set of alternative products or services. Such systems which are usually referred to as *recommender systems* has been used for many different information items such as books, music, films, news, scientific literatures (Werthner et al., 2007).

Knowledge gained by web data representing users' navigational behaviour on the websites can be applied to provide different information and services to website users. Any activities in website design to fulfil this aim is called web personalization (Eirinaki and Vazirgiannis, 2003). Web personalization can be performed for particular users or user segments. The main purpose to web personalization on an e-commerce website might be converting website browsing behaviour into customers. It also can help to improve website design, customer retention and loyalty (Nasraoui, 2005). Mobasher et al. (2000) has outlined a good description of the different kinds of web personalization which can affect essential Customer Relationship Management (CRM) activities.

Customer trust is a key issue in e-commerce. Customers are supposed to receive a secure system when they make online payments. A hacked system and misused data would cause serious problem for a company, as it is hard to regain customers' trust. Web data sources and mining techniques are used for intrusion detection and anomaly detection. Statistical models on web usage data are used to identify attacks and to indicate doubtful activities by an authorized user. Normal transaction behaviour is usually captured by a statistical model and activities are compared to the model (Banks and Said, 2006).

The growth of e-commerce permits the customer to choose from several alternatives. The ability of the customers to check the products of e-vendors working in the same trade by moving from a website to another in a short time has affected customer loyalty. Visitors to an e-commerce website display slight loyalty to the specific website when searching for a product or category (Johnson et al., 2004). Additionally, the percentage of website visits that result in purchasing is very low (Bucklin and Van den Poel, 2003). Consequently, e-vendors need some effective levers to remain in such a competitive community. This persuades e-vendors to use data obtained from web users to discover useful knowledge to help to keep customers loyal to the e-commerce website (Abraham, 2003).

Although, e-commerce provides a considerable amount of data, they do not follow many assumptions which are useful for statistical modelling, like independent observations. Furthermore, the number of people who do shopping online is increasing and more businesses are adding electronic services. It means that e-commerce data is increasing in volume, and it is not easy to extract interpretable descriptions that support business decisions.

Considering the sheer increase of data in size and complexity, more intelligent knowledge mining techniques are necessary (Roussinov and Zhao, 2003; Abraham, 2003). This persuaded researchers to contemplate developing methodologies by which the hidden patterns of web behaviour visits become apparent through exploring web data. This broad class of research is usually referred to as web mining analysis (Chakrabarti, 2003).

1.2 Web Mining

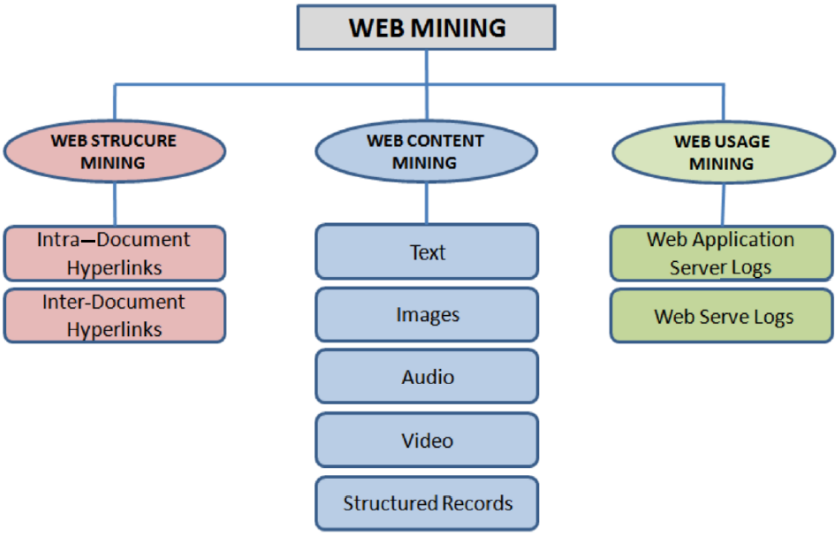
Web Mining includes the application of data mining methodologies, techniques, and models to all kind of data forms relating to the World Wide Web. On the other hand, data mining is the analysis of large data sets to find un-suspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner (Cooley et al., 2000). There are three types of data available in the web, which have been a focus of attention: web usage data, also known as clickstream data; web content data; and web structure data (Cooley et al., 1999a). Therefore, by mining the web, we refer to uncovering patterns in web content, structure, and usage demonstrates by means of data mining methods and models. Accordingly, based on research studies carried on the different web-related data, web mining has been categorized into three domains: content, structure and usage mining (Chakrabarti, 2003). In what follows, we will provide a general overview of all three kinds of web-based data. Figure 1.1 depicts a diagram about the taxonomy of the Web Mining.

1.2.1 Web Usage Mining

Each click made by a user on a web browser while surfing the Internet, corresponding to an HTTP request sent to the server of the website, generates a single entry in the server access logs. Each log entry may includes such information as fields identifying the hitting date and time of the request, HTTP Status, bytes sent, download time, the server IP address of the user, the resource requested, status of the request, HTTP method used, browser and operating system type and version, the referring web resource, and, if available, client-side cookies which uniquely identify a repeat visitor (Johnson et al., 2004). This information varies depending on the log format. The file containing this information, usually referred to as a web log file, is the primary source of data representing the navigational behaviour of visitors.

In addition to web logs, the operational database(s) for the website may contain additional

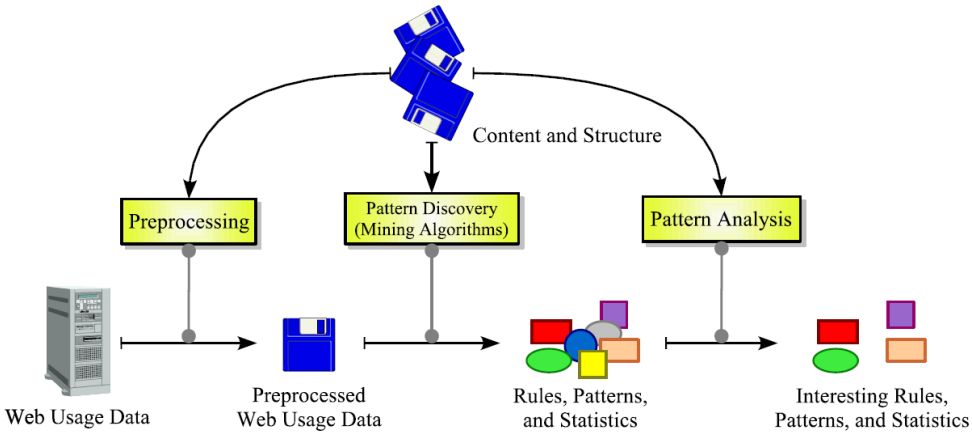
Figure 1.1: *A diagram about web mining taxonomy (Cooley et al., 2000)*



information regarding user profile, user conversion (desired action based on direct requests from marketers, advertisers, and content creators– usually making an online purchase for an e-commerce website), user demographics for registered users, and user ratings on various objects such as products, music, films, past purchases, etc. Some of these data can be captured anonymously as long as it is possible to distinguish between different users (Mobasher, 2006). For example, anonymous information contained in client-side cookies can be considered a part of the user's profile information, and used to identify repeat visitors to a site. E-commerce exploits this data along with clickstream data to have a more clear picture of the user's behaviour. These data may be available on separate servers and will need to be merged with the web logs before preprocessing can be done.

Web usage mining is the application of data mining techniques to large web data repositories in order to extract interesting and useful knowledge and implicit information that reflect the behaviour of humans as they interact with the Internet (Cooley et al., 1999a; Kosala and Blockeel, 2000). Some of the data mining algorithms that are used in web usage mining employ statistical modelling, clustering and classification, association rule generation, and sequential pattern generation (Kosala and Blockeel, 2000).

Figure 1.2: *A web usage mining process (Cooley et al., 2000)*



1.2.2 Web Content Mining

The content data in a site is the collection of objects and relationships that is conveyed to the user. For the most part, this data is comprised of combinations of textual materials and images. The data sources used to deliver or generate this data include static HTML/XML pages, multimedia files, dynamically generated page segments from scripts, and collections of records from the operational databases. The site content data also includes semantic or structural meta–data embedded within the site or individual pages, such as descriptive keywords, document attributes, semantic tags, or HTTP variables (Kolari and Joshi, 2004). The underlying domain ontology for the site is also considered part of the content data. Domain ontologies may include conceptual hierarchies over page contents, such as product categories, explicit representations of semantic content and relationships via an ontology language such as RDF, or a database schema over the data contained in the operational databases (Kosala and Blockeel, 2000). This information is an unstructured data source contrary to fully structured data like database tables. Web content mining techniques are also applied to this unstructured data embedded in web documents and services. Web content mining is sometimes called web text mining, because the text content is the most widely researched area. Most of the focus on web content mining techniques is on clustering and classification analysis.

1.2.3 Web Structure Mining

Web structure data refers to the information available in the inter-page linkage structure among web pages as well as intra-page linkage structure of content within a page. This data represents the designer's view of the content organization within the site. Structure for a site can be revealed by an automatically generated site map. Web Structure Mining discusses basic ideas and techniques for extracting text information from the web, including collecting and indexing web documents and searching and ranking web pages by their textual content and hyper-link structure (Chakrabarti et al., 1998). Note that we may need to have a knowledge of the topology or structure of the website, the network hierarchies and relationships among the web pages to enable us to perform a preprocessing task such as *path completion*.

1.3 Clickstream Data

Clickstream data gathered from a website can provide insight into the behaviour of the website visitors. To illustrate what type of information can be collected while a user is surfing the internet, consider the typical behaviour of a user who decides to purchase a product online. This task starts by signing onto the internet and using a search engine to find what sites sell a favourite brand. Then the user would click on the first link represented by the search engine and begin to browse inside the website. Shopping in the website would result in adding an item to a shopping cart. To make an order, personal information is completed using online forms. This might include credit card number and shipping address. The next page usually displays the order information and total cost. Finally, by confirming the order, one checks whether a confirmation email has been sent. If so, one would sign out of the internet. During this entire process, *clickstream data* has been collected in the web log file of the Web retailer's server (Werner et al., 2002).

The *clickstream* is defined as the aggregate sequence of page visits executed by a particular user as the user navigates through a website. It consists of the records of a user's activity on the internet, including how one got to the website, every website and every page of the website that the user visits, how long the user was on a page or site, in what order the pages were visited, the point at which he left the website, the merchandise he considered buying, any newsgroups that the user participates in and even the e-mail addresses the user provides for correspondence (Mobasher et al., 2000). In the next section we get familiarized with web server log files as a main source of data available for the analysis of users' behaviour on the Web.

1.3.1 Web Server Log Files

For each request from the user's browser (Internet explorer, Mozilla Firefox, Netscape, etc.) to a web server, a response is generated automatically, called a *web log file*, *log file*, or *web logs*. This response takes the form of a simple single-line transaction record that is appended to an ASCII text file on the web server. This text file may be comma-delimited, space-delimited, or tab-delimited. There are two standard Web log formats, The Microsoft standard format (or Microsoft professional Internet services format) and the National Center for Supercomputing (NCSA) common log file format. The field definitions of clickstream data are thus already defined to help simplify and reduce development time when dealing with these formats. As follows, we will illustrate the different fields which exist in a typical log file (Markov and Larose, 2007).

Remote Host Field

This field consists of the Internet Protocol (IP) address of the remote host making the request for the user visiting a website. An IP address is a numerical identification and logical address that is assigned to devices participating in a computer network utilizing the internet protocol for communication between its nodes (Comer, 2000). IP addresses are stored in binary numbers, but they are usually displayed in human readable notations. An IP address comprises three pieces of information: a name, an address, and a route. The name indicates what a visitor seeks. The address helps to find out where it is, and the route shows how to get to the address. When the remote host name is not available, then the domain name system (DNS) can help to decode the host names into IP address and vice versa. In view of the fact that humans prefer to work with domain names and computer are most efficient with IP addresses, the DNS provides an important interface between human and computer (for more information see the Internet Systems Consortium, www.isc.org). IP addresses are usually represented by dot-decimal notation, four numbers each running from 0 to 255. A typical user IP address in the log entries would be `141.243.1.172`.

Identification Field

Identification fields show the login IDs of users who have entered a password protected area of the site. This field is used to store identity information by the client only if the web server performs an identity check. However, this field is rarely used because identification information provided is in the form of text rather than a securely encrypted form. Therefore, this field usually contains a $\sim$, or $--$, indicating a null value (Markov

and Larose, 2007; Liu, 2006).

Date and Time Fields

The date and time of the local server for each page request is recorded in the log file in Greenwich Mean Time (GMT). It is more common for the date/time field to follow the following format: `dd/Mon/yyyy:hh:mm:ss offset`, where `hh:mm:ss` represents 24–hour time, given in Eastern Date Time (EDT) system, `dd/Mon/yyyy` represents the date, and an `offset` is a positive or negative constant indicating in hours how far ahead of or behind the local server is from Greenwich Mean Time (GMT). For example, a date/time field of `12/Sep/2008:13:21:02 -0700` indicates that a request was made to a server at 01:21 p.m. on September 12, 2008, and the server is 7 hours behind GMT.

HTTP Request Field

The HTTP request field consists of the information requested by the user's browser from the web server. The entire HTTP request field is embraced within quotation marks. Basically, this field can be partitioned into four areas: the request method, Uniform Resource Identifier (URI), the header, and the protocol. The most common request method is `GET`, which represents a request to retrieve data, identified by URI. For example, `GET/index.html HTTP/1.1` represents a request from the user's browser for the web server to provide the webpage `index.html`. Besides GET, other requests include HEAD, PUT, and POST. For more information about these request methods, refer to the W3C World Wide Web consortium, www.w3.org. The URI contains the page or document name and the directory path requested by the client browser. It may also contain information concerning the *keywords* are being used by user in search engines that point to the website. The keyword are terms and phrases that can be used to find the relevant link by a search engine. The HTTP request field also includes the protocol section. This indicates which version of the HyperText Transfer Protocol is being used by the clients browser, `HTTP` in the mentioned example. Based on the relative frequency of newer protocol version (e.g., `HTTP/1.1`), the web developer may decide to take advantage of the greater functionality of the newer version and provide more online features.

Referrer Field

The referrer field provides information about the webpage that the user came from. In this case, the referrer field lists the URL of the previous website visited by the client, which linked to the current page. For images, the referrer is the webpage on which the

image is to be displayed. The referrer field contains important information for marketing purposes, since it can track how people found a website. Again, if the information is missing, a dash is used.

User Agent Field

The user agent field indicates the user's browser, browser version, and operating system. This field can also contains information regarding bots or *web crawlers*. A Web crawler, or *bot*, is a computer program that browses the internet in a automated manner to provide up-to-date data for a specific purpose. A bot may be used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages, to make a faster search. It also can be used to collect e-mail addresses, usually for sending spam. A web developer can use user agent field to block certain sections of the website from the web crawlers, in the interests of preserving bandwidth. This field also enables the analyst to determine whether a human or a bot has accessed the website, and thereby to omit the bot's visit from the analysis.

Status Code Field

Once a browser request fails, a three digit response from the web server is transmitted to the user, and recorded in the web log file. This field, referred to as a status code, indicates whether the request was successful, or if there was an error. In the case of error, it also indicates which type of error occurred. Codes of the form `2xx` indicate that the request from the client was received, understood, and completed. Codes of the form `3xx` indicate that further action is required to complete the client's request. Codes of the form `4xx` are used to show that the client's request cannot be fulfilled, due to incorrect syntax or a missing file. Finally, codes of the form `5xx` indicate the failure of the web server to fulfil what was apparently a valid request (Markov and Larose, 2007).

Transfer Volume Fields

The transfer volume field indicates the size of the file, in bytes, sent by the web server to the client's web server. Only `GET` successful requests, status code = 200, will have a positive value in the transfer volume field. Otherwise, it will consist of a dash or a value of zero. This field is useful to monitor network traffic, the load carried by the network through a 24–hour cycle.

Figure 1.3 depicts a fragment (three entries) of a log entry of a typical web server log file. For the first entry, it shows a user with IP address `175.12.131.24` accessing a

Figure 1.3: *A fragment of a common server log file*

```
2009-02-01 10:08:43 175.12.131.24 - GET /wiki/Clickstream - 200 9221 HTTP/1.1
en.wikipedia.org Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+2.0.50727)
http://www.dur.ac.uk/
2009-02-01 10:08:45 212.137.91.10 - GET /wiki/Clickstream - 200 9221 HTTP/1.0
en.wikipedia.org Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+2.0.50727)
http://www.yahoo.co.uk/
2009-02-01 10:09:4 175.12.131.24 - GET /wiki/Clickstream - 200 9221 HTTP/1.1
en.wikipedia.org Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+2.0.50727)
http://www.dur.ac.uk/
```

resource `/wiki/Clickstream` on the server `en.wikipedia.org` at `2009-02-01 10:08:43` in Greenwich Mean Time. The browser type (Microsoft Internet Explorer) and version (6.0), as well as operating system information on the client machine (Windows NT) are captured in the agent field of the entry. Finally, the referrer field specifies that the user came to this location from an outside source: `www.dur.ac.uk`. The second entry has a different ISP, consequently it confirms that this records was produced by a different user. The third record is from the first user, with the same ISP, but the URL has changed to a new page.

1.3.2 Preprocessing

The web log files are not well-structured data and cannot be directly used for web mining purposes. For example, when a user requests a web page containing graphic and sound files, the request results in several records/lines in the web log file that represent just one page request. One also need to remove the records in the web log files which are made by bots, as those line do not reflect the human browsing behaviour. Considering the irrelevant information in web logs files, from the web usage mining point of view, it is required as an essential data preparation activity to convert the raw data into data abstraction necessary for further analysis (Natheer and Chan, 2006; Helmy et al., 2008), usually referred to as a *data preprocessing* step. Cooley et al. (1999a) provides a comprehensive discussion of the stages and tasks in data preparation for Web usage mining. In this section we briefly illustrate some common preprocessing task.

User Identification

Since a user may visit a site more than once, before modelling user behaviour one needs to distinguish between different users. There is no emphasis to obtain knowledge about

user identity, but one needs the sequence of activities performed by the same user during different sessions, which is usually called *user activity*.

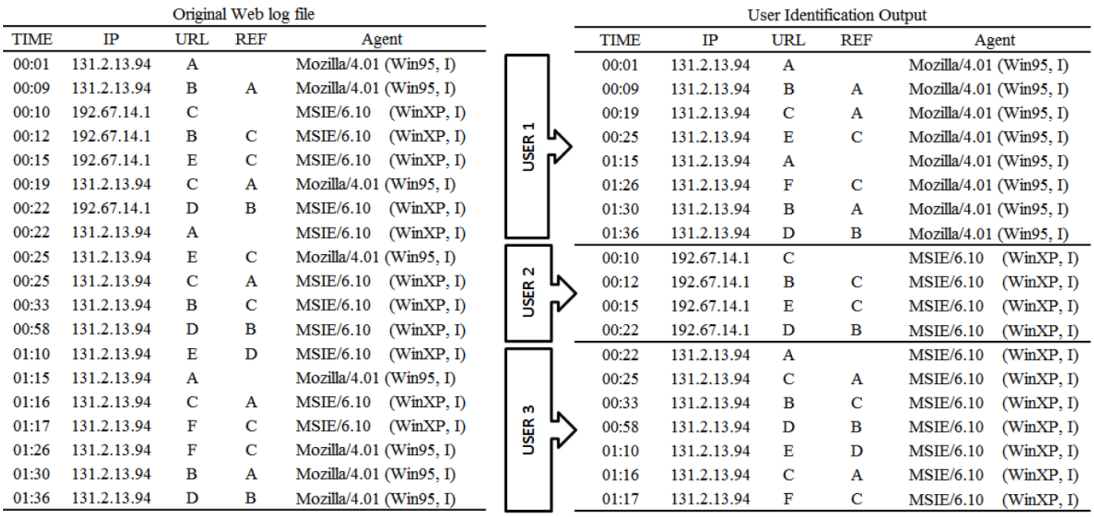
The most reliable way to identify users is by user registration. In this case, each user has his/her own user ID for logging into the website. It has also the advantage of collecting additional demographic information about users. Unfortunately, due to privacy concerns, many users are not interested in browsing in a website when they are asked for registration and logins. Even for registered users, many prefer to provide false information (Cooley et al., 1999a).

Another way to identify users is based on the IP address, as users with different IP address are definitely are people who connected from different computer. Due to the increasing number of internet users, ISP proxy servers rotate IP addresses which are assigned to clients as they browse the Web. Therefore, one may find many identical IP addresses in log entries, due to the limited number of proxy server IP addresses, from large internet service providers (Mobasher, 2007). It obliges us to find a way to distinguish among those who come to the website with the same IP address. One heuristic is to use a combination of IP address with other clickstream information such as user agent field or referrer. It can be assumed that each different agent type, including browser software, or its version, or operating system for an IP address represents a different user (Pirolli et al., 1996). One may take advantage of site topology to construct browsing paths for each user. Afterwords, if a page is requested that is not directly accessible by a hyper-link from any of the pages visited by the user, again, it can be assumed that there is another user with the same IP address (Liu, 2006).

Note that user identification on log server data actually helps to distinguish between machines rather than users, except in the case of registered users of a website who log into the website through a user ID and a password. For example, if a user visits a website from a machine in the work office, and later returns to the website from home, the user identification pre-processing fails to identify the user. Oppositely, when a machine is used by several users, browsing the website by different people might be considered as a re-visiting of the website by the same user.

Figure 1.4 depict a fragment of a typical web log data file, Using a combination of IP and agent fields in the log file, we are able to partition the log into activity records for three separate users, as depicted on the right panel. The IP is used first, and the different IP addresses get separated. In the next step the agents of the each group of web log entries with the same IP are investigated to find whether all the agents are the same. The entries with the same log but dissimilar agent are separated as different Users.

Figure 1.4: *A sample of user identification using IP and agent fields, adopted from (Liu, 2007)*



Original Web log file

TIME	IP	URL	REF	Agent
00:01	131.2.13.94	A		Mozilla/4.01 (Win95, I)
00:09	131.2.13.94	B	A	Mozilla/4.01 (Win95, I)
00:10	192.67.14.1	C		MSIE/6.10 (WinXP, I)
00:12	192.67.14.1	B	C	MSIE/6.10 (WinXP, I)
00:15	192.67.14.1	E	C	MSIE/6.10 (WinXP, I)
00:19	131.2.13.94	C	A	Mozilla/4.01 (Win95, I)
00:22	192.67.14.1	D	B	MSIE/6.10 (WinXP, I)
00:22	131.2.13.94	A		MSIE/6.10 (WinXP, I)
00:25	131.2.13.94	E	C	Mozilla/4.01 (Win95, I)
00:25	131.2.13.94	C	A	MSIE/6.10 (WinXP, I)
00:33	131.2.13.94	B	C	MSIE/6.10 (WinXP, I)
00:58	131.2.13.94	D	B	MSIE/6.10 (WinXP, I)
01:10	131.2.13.94	E	D	MSIE/6.10 (WinXP, I)
01:15	131.2.13.94	A		Mozilla/4.01 (Win95, I)
01:16	131.2.13.94	C	A	MSIE/6.10 (WinXP, I)
01:17	131.2.13.94	F	C	MSIE/6.10 (WinXP, I)
01:26	131.2.13.94	F	C	Mozilla/4.01 (Win95, I)
01:30	131.2.13.94	B	A	Mozilla/4.01 (Win95, I)
01:36	131.2.13.94	D	B	Mozilla/4.01 (Win95, I)

User Identification Output

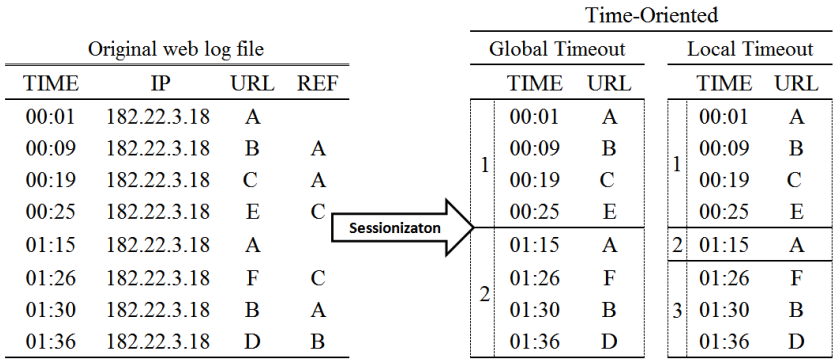
TIME	IP	URL	REF	Agent
00:01	131.2.13.94	A		Mozilla/4.01 (Win95, I)
00:09	131.2.13.94	B	A	Mozilla/4.01 (Win95, I)
00:19	131.2.13.94	C	A	Mozilla/4.01 (Win95, I)
00:25	131.2.13.94	E	C	Mozilla/4.01 (Win95, I)
01:15	131.2.13.94	A		Mozilla/4.01 (Win95, I)
01:26	131.2.13.94	F	C	Mozilla/4.01 (Win95, I)
01:30	131.2.13.94	B	A	Mozilla/4.01 (Win95, I)
01:36	131.2.13.94	D	B	Mozilla/4.01 (Win95, I)
00:10	192.67.14.1	C		MSIE/6.10 (WinXP, I)
00:12	192.67.14.1	B	C	MSIE/6.10 (WinXP, I)
00:15	192.67.14.1	E	C	MSIE/6.10 (WinXP, I)
00:22	192.67.14.1	D	B	MSIE/6.10 (WinXP, I)
00:22	131.2.13.94	A		MSIE/6.10 (WinXP, I)
00:25	131.2.13.94	C	A	MSIE/6.10 (WinXP, I)
00:33	131.2.13.94	B	C	MSIE/6.10 (WinXP, I)
00:58	131.2.13.94	D	B	MSIE/6.10 (WinXP, I)
01:10	131.2.13.94	E	D	MSIE/6.10 (WinXP, I)
01:16	131.2.13.94	C	A	MSIE/6.10 (WinXP, I)
01:17	131.2.13.94	F	C	MSIE/6.10 (WinXP, I)

Sessionization

The purpose of session identification is to divide the page accesses of each user from the clickstream data into an actual sequence of single user navigations during a single visit to the website. Session identification, which is sometimes named as episode identification, is usually performed as a final step in preprocessing of the clickstream data (Markov and Larose, 2007). In the lack of additional authentication information from users and without mechanisms such as embedded session ids, sessionization is carried out based on heuristic methods. Various heuristics for sessionization have been introduced and studied by Cooley et al. (1999a). More recently, Spiliopoulou et al. (2003) proposed a comprehensive framework for evaluating the effectiveness of such heuristics, and the impact of different heuristics on various Web usage mining tasks has been analysed by Berendt et al. (2002)

Sessionization heuristics are categorized into two basic groups: time-oriented and navigation-oriented. The time-oriented heuristic applies time-out estimates to distinguish between successive sessions. For logs with long periods of time, it is very likely that users visit the website more than once. One heuristic is to assume that the user starts a new session, whenever the time between page requests exceeds a certain limit (Cooley et al., 1999a). As an example of a time-oriented heuristic, one may scan the user activity log and partition it into different sessions whenever the total session duration exceeds a threshold θ. One may choose the total time spent between two subsequent requests and split the session when it exceeds a threshold δ. It is usual to take 30 minutes threshold as a default

Figure 1.5: *A Sample for sessionization based on global time threshold $\theta = 30$ minutes and local time threshold , $\delta = 10$, minutes (Liu, 2006)*

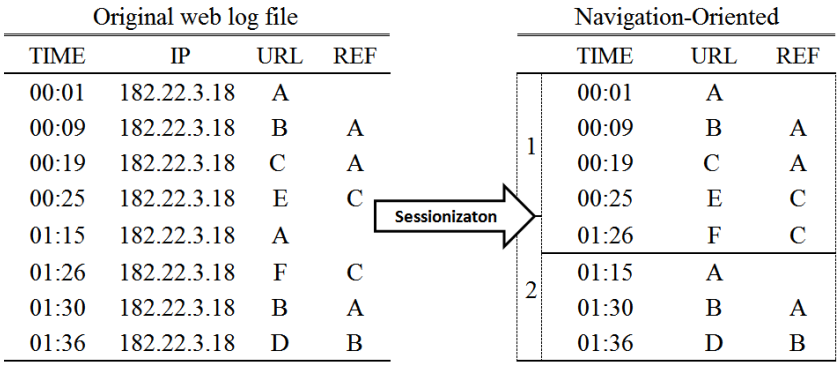


time-out, and Catledge and Pitkow (1995) established a time-out of 25.5 minutes based on empirical data. It would be more efficient to find an appropriate time-out after analysing the web logs, and to use different settings for each website (Liu, 2006).

Navigation-oriented sessionization uses either the static site structure or the implicit linkage structure captured in the referrer fields of the server logs (Cooley et al., 1999a). A common way is to assign a request to a constructed session if the referrer for that request was previously invoked in the session. Otherwise, the request is considered as the start of a new session. Note that a request may have been accessed previously in multiple sessions. In this case, the request belongs to more than one open constructed session. One may use additional information to assign the request properly. For example, the request could be added to the most recently opened session satisfying the above condition (Liu, 2006).

An example of the application of sessionization heuristics is given in Figure 1.5. Applying a global time threshold with $\theta = 30$ minutes, the user activity record has been partitioned into two separate sessions. If we were to apply a local time threshold of $\delta = 10$ minutes, the user record would be seen as three sessions, namely, $A \rightarrow$B$\rightarrow$C$\rightarrow$E, A, and F$\rightarrow$B$\rightarrow$D. On the other hand, using the navigational-oriented heuristic on the same user activity record would result in different sessions (see Figure 1.6). once the request for F (with time stamp 1:26) is reached, there are two open sessions, namely, A$\rightarrow$B$\rightarrow$C$\rightarrow$E and A. But F is added to the first because its referrer, C, was invoked in the first session. The request for B (with time stamp 1:30) may potentially belong to both open sessions, since its referrer, A, is invoked both in the first session and in the second session. In this case, it is added to the second session, since it is the most recently opened session.

Figure 1.6: *A Sample for sessionization based on the navigation-oriented approach (Liu, 2006)*



Data Fusion/Merging

Data fusion is an essential preprocessing task on clickstream data where the analysis of user behaviour is performed over the log files of multiple related websites. In large-scale websites, multiple Web or application servers are used to show the content served to the users. The web log files produced by different servers need to be merged properly to capture the users' trace. Fusion is followed by the sessionization and user identification preprocessing methods, in combination with the heuristics based on the *referrer* field in the web log file. Tanasa and Trousse (2004) introduce the heuristics to be applied for merging web log data from different servers.

Data Filtering

When loading a particular web page, the browser also requests all the objects embedded in the page. It leads to the registration of several record lines in the web log file for logs, cookies, graphic files, etc. Data filtering involves the task of removing extraneous references to embedded objects that may not be important for the purpose of analysis, including references to style files, graphics, or sound files, which belong to top/bottom frames (Markov and Larose, 2007). The filtering process also involves the removal of some of the data fields, such as number of bytes transferred or the version of HTTP protocol used, that may not provide useful information in analysis or data mining tasks (Werner et al., 2002). In addition to these fields, we may also omit some web log entries in processing to quickly identify the exact records that we need from the Web logs. One example is the removal of the log entries associated to the users which have sent just one request. These single-page-visit users are usually referred to as users who have found the

website irrelevant to their needs and leave the website quickly.

Filtering can be accomplished by checking the suffix of the URL name. For example, all log entries with filename suffixes such as, `gif`, `jpeg`, `GIF`, `JPEG`, `jpg`, `JPG`, and `map` can be removed. In addition, common scripts such as `count.cgi` can also be removed. Web analytic systems use a default list of suffixes to remove files. Note that data filtering is usually site-specific, so the list can be modified depending on the type of site being analysed. For example, suppose we aim to analyse user behaviour for a website that contains a graphical archive. In this case, the log entries of graphics files represent explicit user actions, and should be retained for analysis. Therefore, filtering requires keeping all of the `GIF` or `JPEG` files from the server log file (Markov and Larose, 2007).

Despidering

With information overload on the web, web search engines start the task of gathering information on the web and provide relevant web links containing the information needs of customers. This task is performed by dispatching automatic programs, usually called spider, crawler, or web bots, which browse all over the web and gather information for the search engine databases. The behaviour of bots differs to human behaviour, as they usually request all possible links of the website one after the other. In fact, for using clickstream data to capture web usage, we need to remove this type of non-human access behaviour. Despidering refers to the action of removing references from web log file due to crawler navigations. With the growth of search engines and internet usage, it is likely to find a considerable percentage of references resulting from search engines, spiders, or other crawlers (Markov and Larose, 2007).

Famous search engine crawlers (such as Google, msn, Yahoo!, etc.) can be identified by checking the user agent field of the web log entries. Maintaining a list of such bots, one can remove all records of the web log when its referrer belongs to the list. Another heuristic to detect crawlers is to inspect the typical behaviour that crawlers may follow. For example, *Well-behaved* crawlers that respect the standard robot exclusion protocols strive to access an exclusion file named `robot.txt` in the server root directory in the first attempts of crawling. As a result, One may identify such crawlers by detecting sessions that begin with access to this file. However, for those crawlers that either do not identify themselves explicitly or implicitly; or those crawlers that deliberately masquerade as real human users, identification and removal of crawler references may require the more complex heuristic methods or anomaly detection techniques. For example, Tan and Kumar (2002) applies classification algorithms to build models of crawlers and Web robot navigations.

Path Completion

Path completion refers to the task of filling in page references that are not recorded in the web log file, due to browser and proxy server caching. When a user returns to a page that has been already visited (downloaded) during the same session, the second access to that page will result in viewing the previously downloaded version of the page without sending any request to the server. This lack of request from the browser results in a missing reference in the web log file. Missing references due to caching can be heuristically inferred through path completion which relies on knowledge of the site structure and referrer information from server logs (Cooley et al., 1999a).

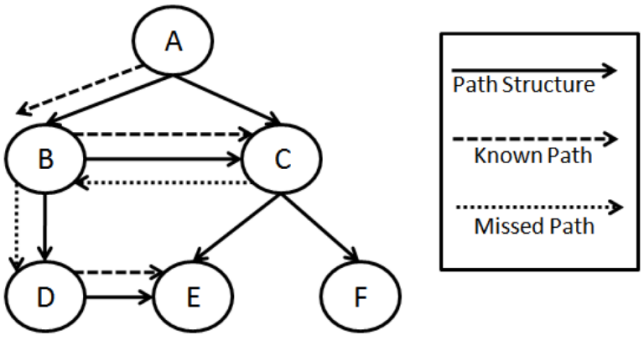
If a page request is not directly linked to the last page a user requested, one can check the referrer log to see if the page is in the user's recent request history. If so, one can assume that the user backtracked with the back button. Otherwise, the site topology must be used to the same effect. If more than one page in the user's history contains a link to the requested page, a reasonable option is to take the closest page to the previously requested page. It is also required to estimate the time of each added page reference. An approach is to assume that any visit to a page already seen makes it work as an auxiliary page, which is used to guide the user to the new pages. The average reference length time (the average amount of time spent on each page) for auxiliary pages of the website can be used as an estimate of the access time for the missing pages.

For example consider a website topology given in Figure 1.7. Let suppose a typical user whose navigational path in the log file is presented by A→B→C→D→E (depicted by grid lines in the Figure 1.7). Since there is no link from C to D, it seems obvious that the user has backtracked, using the browser's *back* button, to page B and then D and E. Due to using a cached file on the client-side, the back reference from C to B does not appear in the log file. Given site structure and the referrer information, we infer the missing references C→B and B→D (depicted by a dashed line in Figure 1.7). The path completion step results in actual user paths of A→B→C→B→D→E.

Data Integration

An effective framework for knowledge discovery in e-commerce is usually performed by integrating the preprocessed clickstream data with user data from different sources. Online purchase, which is usually called a *conversion* in this context, is of the highest importance regarding integration with clickstream behaviour. E-vendors are interested to find patterns of behaviour between people who purchase online and those who do not. Other

Figure 1.7: *A Sample for path completion by diagram of the website structure. The navigational path represented by log file and the missed path is depicted by different kinds of arrows (Liu, 2006)*



user data such as demographics and purchase histories, especially in the case of registered users, also provide valuable information in pattern discovery. Operational databases may present information about product categories and attributes (Kohavi et al., 2004). Part of the data to integrate with clickstreams are produced by events which shows a user's tendency to buy the products presented in the website. For example, shopping cart changes, order information, impressions (the action of the user to visit a page containing an item of interest), click-through (the action of the user to click on an item of interest in the current page), provide additional data about users surfing the website (Kimball and Merz, 2000). The integrated database enables web managers to produce e-metrics including frequency of purchases, the value of purchases, the number of different items purchased, the number of different item categories purchased, average time spent on pages or sections of the website, day of week and time of day for certain activities, response to online recommendations specials, etc. (Buchner and Mulvenna, 1999).

Pageview/Transaction Identification

For a static single frame site, each page request in clickstream data corresponds to a specific user action. For example, clicking on a link, reading an article, viewing a product page, adding a product to the shopping cart, or visiting the index page, etc. These actions result in a collection of web objects or resources generated by the user's web browser. The task of transforming and aggregating semantically meaningful user groupings of page references is called *pageview* (or *transaction*) *identification* (Cooley et al., 1999a). Depending on the goals of the analysis, this data aggregation is performed at different levels of abstraction. The most basic level of data abstraction is that of all webpages. How-

ever, it may be desirable to consider pageviews at a higher level of aggregation, where each pageview represents a collection of pages or objects, for example, pages related to the same concept category (Liu, 2006). Transaction identification also depends on the navigational structure of the site, as well as on the page contents and the underlying site domain knowledge. For example, Moe (2003) used the general content of pages viewed to categorize pages as buying, browsing, searching or knowledge-building pageviews.

In e-commerce websites, pageviews may correspond to various product-oriented events, such as product views, registration, shopping cart changes, purchases, etc. In this case, identification of pageviews may require a priori specification of an event model based on which various user actions can be categorized.

1.3.3 Data Structures

The web log data has the potential to provide a data structure in which records represent a single visit to a Website, for users who accessed the website, and fields comprise corresponding attributes of the the website being visited. The information includes the webpages requested, as well as the order of the webpages, the amount of time spent on each page, and any other possible derivatives from the web log file entries. This data is usually referred to as user-session data, in view of the fact that its records represent the sessions of website viewing. As a user may visit a website several times, it is likely to have multiple sessions (records) corresponding to an individual user. In fact, the output of session identification makes a set of sessions $S = \{S_1, S_2, \ldots, S_m\}$ which are uniquely marked by session ID field. The attributes of the session can be extracted from the web log file, including information such as the date of the session, whether the session is a weekend session, total time duration, number of web pages visited, whether the referrer is of search engines, etc. User identification provides a set of users, $U = \{U_1, U_2, \ldots, U_n\}$ and corresponding user ID fields. The user identifier helps to establish whether the website has already been visited by the user in a specific session. Therefore, another session attribute is produced by categorizing the session as a repeat session or a first-visit session. Figure 1.8 shows part of a typical user-session data set

For some data mining tasks, such as clustering and association rule mining, where the ordering of pageviews in a session is not relevant, The user session is represented as a vector over of size k of pageviews, $P = (P_1, P_2, \ldots, P_k)$ which is a result of a pageview identification process. As mentioned earlier, pageviews can be webpage categories (or product page entities) to which mining tasks are applied. Ignoring the order of pages requested, each of the pageview categories can be considered as a character of a session

Figure 1.8: *A Sample of user-session data produced by the web log file after data preprocessing (in some literature this is called a transaction matrix)*

SID	UID	VH	TD	VW	NPV	RS
1757374	35477	9	467	1	10	0
1757375	35477	12	34	1	2	1
1757376	35477	10	243	0	3	1
1757377	35478	23	1546	0	20	0
1757378	35478	0	865	0	12	1
1757379	35478	23	120	0	2	1
1757380	35479	13	490	0	3	0
1757381	35479	10	830	1	18	1

SID	Session ID
UID	User ID
VH	Visiting Hour
TD	Time Duration
VW	Visiting on Weekends
NPV	Number of Pages Visited
RS	Repeat Session

and represented by a field in user-session data, in some literature called *user-pageview* data (Liu, 2006). The fields contain the weight, representing its significance, associated with the pageview in the session. The weights can be determined in a number of ways, in part based on the analysis purpose and type. In most of the analysis tasks the weights are binary, representing the existence or non-existence of a pageview in the session (Mobasher, 2007). In some websites where users are asked to rank the items in the webpages, weights may be based on user ratings of items. A reasonable weight can be assigned by a function of the duration of the pageview in the user's session. As the time spent by a user on the last pageview in the session is not available, a heuristic is to set the weight for the last pageview to be the mean time duration for the page taken across all sessions in which the pageview does not occur as the last one. Using the normalized value of page duration instead of raw time duration is recommended in order to take into consideration variations in session time duration. In applications, the log of pageview duration is used as the weight to reduce the long tail of distribution noise in the data. A sample pageview data set has been shown in Figure 1.9. The value associated to each page for the user is the total time spent by the user on the page.

The ordering of pageviews visited by users will also contain information about their browsing behaviour. A web manager might be interested in analysing the clickstream path taken by users during their session on the website (Berkhin et al., 2001). Furthermore, clustering of users can be implemented based on methods which consider the ordering

Figure 1.9: *A typical pageview data set, or pageview part of a user-session file, representing pageview attributes (total time spent on the page in this case) associated to the session*

SID	UID	A.html	B.html	C.html	D.html	E.html
1757374	35477	9	77	3	0	0
1757375	35477	12	36	0	14	0
1757376	35477	45	77	12	0	0
1757377	35478	120	29	18	12	0
1757378	35478	3	56	0	55	19
1757379	35478	14	16	0	43	5
1757380	35479	98	12	43	3	0
1757381	35479	5	88	14	34	0

of pageviews, for example model-based clustering by means of first-order hidden Markov models (Smyth, 1997; Ypma and Heskes, 2002). Therefore, it is necessary to prepare a data structure which represents the sequence of pageviews. The sequence of pageviews for different sessions can be represented by vectors of different lengths for each session, $\boldsymbol{j} = (j_1, j_2, \ldots, j_m)$, where $m \geq 1$. Corresponding to the sequence of the pageviews visited, the index of the pageviews in the state space $P = \{P_1, P_2, \ldots, P_k\}$ is used to represent the sequence. For example, a vector of $\boldsymbol{j} = (1, 4, 1, 2, 6)$ shows the pageviews of $P_1 - P_4 - P_1 - P_2 - P_6$ respectively during the session. One might apply weighted analysis of the sequences by considering a sequence of weights joint to the pageview sequence, $\boldsymbol{w_j} = (w_{j_1}, w_{j_2}, \ldots, w_{j_m})$, for example based on the amount of time spent on each pageview.

It is also possible to integrate the conversion file with the user-session data set. The conversion file comprises the information about online purchases. This file helps to add a binary attribute of whether a session resulted in an online purchase of one or more products/services, the amount of purchase, as well as codes of the product/services. The integrated data set allows the analyst to investigate the browsing behaviour that has value to the business, usually an online sale in-commerce websites (Van den Poel and Buckinx, 2005). This type of analysis is called *conversion analysis*. It also enables the analyst to inspect the impact of marketing campaigns, such as email, banner, referral and custom defined marketing campaigns.

For registered users who login to the website, depending on the requested information in the registration forms, there is more information available about the user. This includes attributes such as gender, age, occupation categories, educational level, etc. The provision of information in registration forms, information such as telephone number, age

and income, can be used to construct a measure of *trust* concerning customers (Van den Poel and Buckinx, 2005). However, due to privacy concerns, some people tend not to give correct answers to profile questions. Registration gives the web owner access to the history of purchase behaviour and the interests of the users (Moe and Fader, 2004). Historic purchase behaviour is already proved to be commonly used in analysing customer behaviour in offline settings (Schmittlein and Peterson, 1994). This may also provide a measure loyalty to the website, as the amount a customer is buying is commonly used as an indication of loyalty (Jones and Sasser, 1995).

Generally, the textual features from the content of web pages represent the underlying semantics of the site. Each pageview can be represented as an r-dimensional feature vector, $F = (F_1, F_2, \ldots, F_r)$, where r is the total number of extracted features (words or concepts) from the website vocabulary list. For each pageview j there exist $F^j = (F_1^j, F_2^j, \ldots, F_r^j)$, where F_i^j is the weight of the ith feature in pageview j, for $1 \leq j \leq r$. For the whole collection of pageviews in the site, we then have a $k \times r$ pageview-feature matrix. The integration process may, for example, involve the transformation of user transactions (in user-pageview matrix) into *content–enhanced* transactions containing the semantic features of the pageviews. The goal of this is to represent each user session (or more generally, each user profile) as a vector of textual (semantic) features or concept labels rather than as a vector over pageviews. In this way, a user's session reflects not only the pages visited, but also the significance of various concepts or context features that are relevant to the user's interaction. We do not employ content data in this thesis.

1.4 SLC User-Session Data

Within this thesis we use data provided to us by a SAYU company. This is clickstream data from commercial websites selling products and services on the internet.We will refer to this data as the SLC data set throughout this thesis. SLC data is a user-session data set comprising two source of information: Server log files and conversion data. As we will not access the registered user data, demographic information about the visitors is not available. Tracking data contains general clickstream information from the website which is obtained using log server files. We also use conversion data which comprises information about visitors who do an online shopping during a web session. The conversion information involves an identifier for conversion records, time, date, IP, agent, and amount of conversion. The total data available to use was collected from May 25th 2008 to June 18th 2008, but to reduce the data size to feasible amounts, we only exploit about one week of clickstream data. The SLC data consists of 10496 records, each record

corresponds to a single visit to the website. This number of visit has been made by 10091 distinct users/machines.

The main preprocessing tasks on the web log data files were implemented by the local company's expert. The preprocessing includes eliminating bots from the web log files (despidering); eliminating irrelevant elements in the log files (filtering), and generating the IDs to enable us to identify users (user identification). We also perform a sessionization by splitting the sessions where the time stamp between two consequent page request lasts more than 30 minutes (Catledge and Pitkow, 1995). The date and time stamp are used to extract any other related to the time or date (e.g., visit on week day, holidays, day time of the session, etc.). Records of the sessions of only one page request (single-page visits) were filtered out, considering them as users who enter the website by mistake (Bucklin and Sismeiro, 2003).

After preprocessing clickstream data, several different measures can be extracted to represent attributes of the website, visit/session, or user. The website attributes which are most often incorporated into web analytic tools for websites includes Website traffic, stickiness/slipperiness, profitability, as well as traversal paths. Clickstream data can provides key performance indicators (KPI) that reflect critical success factors of the website. At the level of sessions, it is highly important to measure the attributes such as frequency, recency, and depth of visit, along with path navigations. For example, the amount of time spent at website during a session which can be served as a measure of depth of visit has been found crucial in understanding the web visitor's behaviour (Padmanabhan et al., 2001). In the next chapter a wider range of the metrics and reports that can be obtained using clickstream data will be discussed in more details.

1.5 Thesis Outline

This thesis is concerned with providing further statistical development in the area of web usage analysis to explore web browsing behaviour patterns. The SLC data set will be used to illustrate the results throughout the reminder of the thesis. The structure of the thesis is as follows:

Chapter 2 provides a brief over-review of the metrics/measures that can be obtained using clickstream data. The metrics will be given at two levels: metrics for the website, and metrics for a web session. These measures will be calculated for the SLC data set and some of the results are reported through tables and graphs. Chapter 3 is devoted to the explanatory analysis of information available in clickstream data. We also take a look at

the some of questions may be of interest when analysing web browsing behaviour. Chapter 4 provides an introduction to the two new robust measure of effect size for Non-normal situation, as some key variables of clickstream data do not follow the Normal distribution.

In chapter 5 our focus will be on conversion analysis, to investigate the causal relationship between the general clickstream information and online purchasing using a logistic regression approach. The aim is to find a classifier by assigning the probability of the event of online shopping in an e-commerce website. We implement an automatic stepwise model selection, we chose the best model, in terms of AIC, by choosing from the main effect model, as well as all possible interaction terms. In comparison with previous studies, our contribution has been to take into account interaction terms, as well as main effect general clickstream information. Our aim was to identify the most significant predictors of online purchasing to maximize the predictive power of our model in practice. We also Compare the predictive performance of the logit model with classification analysis and regression trees (CART).

In chapter 6 we review the theory behind the mixture of hidden Markov models (MixHMM). The parameters estimation will be reviewed and we also extend the EM algorithm for computing the parameters of the MixHMM when observations come from the Poisson, Binomial, Exponential, and the Normal distributions. Then, the Bayesian approach will be developed for MixHMM. We also test this model using a simulated data set. We address the slow mixing problem of the Gibbs sampling, and illustrate how a stochastic forward-backward recursion help improve the mixing of the chain compared to the direct Gibbs sampling algorithm. The performance of the model was assessed over an artificial navigation pattern.

In chapter 7 we develop the application of a MixHMMs to model web browsing behaviour using sequences of web-pages viewed by users of an e-commerce website. The novel contribution of this chapter is to extend the MixHMM in the Bayesian framework in the web data context. This model provides a better understanding of the web browsing pattern in a website. It also can be used to cluster users based on their sequence of web pages visited, where each mixture components represent a class of browsing behaviour. Another practical feature of the model is the ability to make a soft classification of the web pages.

Chapter 2

Metrics and Reports Using Clickstream Data

In this chapter we aim to explain the metrics/measures that can be obtained using clickstream data. These measures are calculated for the SLC data set and some of the results are reported in tables and graphs. It should be noted that we do not discuss all possible web analytical tools which can be used with clickstream data sets, as that is beyond the scope of this chapter. Moreover, the web analyst does not necessarily measure all possible metrics, but is required to calculate measures based on the goal of the website which may vary from site to site. Some useful information regarding web analytics can be found in Kaushik (2007, 2010), Peterson (2004), and Sterne (2002).

2.1 Introduction

Clickstream data has been used to measure different aspects of a website, or behaviour of visitors to a website. In a commercial context it is especially helpful to investigate whether the website works towards the business objectives, by providing key performance indicators (KPI) that reflect the critical success factors of the website, such as website traffic, engagement, and profitability.

Web analytic reports from clickstream data using descriptive statistical methods may include information such as most frequently accessed pages, average length of visit to a page, average length of a specific path through a site, common entry and exit pages, the rate of visits with online purchase, etc. Despite the lack of depth in such reports, the resulting knowledge can be potentially useful for improving the websites performance

and, specifically, providing support for marketing decisions in the case of e-commerce websites. Nowadays, there are many commercial web analytics solutions and services available for analysis of clickstream data including free vendors such as Google Analytics and Yahoo Web Analytics; medium-level paid web analytics services, such as VisiStat, HitsLink and Web-Stat or, notably, enterprise-level services such as Omniture, Oremetrics and Webtrends.

Clickstream data also helps to measure the attributes of web sessions. The frequency, recency, depth of visits, along with path navigations of the visit, all can be obtained using clickstream data (Padmanabhan et al., 2001). The variable most often used in the literature is the number of pages visited in a session (Van den Poel and Buckinx, 2005). The recency, the time elapsed since the last visit, has also been introduced in browsing behaviour studies to understand more about customers virtual shopping habits (Moe and Fader, 2004; Van den Poel and Buckinx, 2005). The time visitors spend on a specific website or a webpage during a session has been found to be crucial in understanding the web visitor's behaviour (Padmanabhan et al., 2001). Bucklin et al. (2002) studied how the number of pages viewed influences visitors' propensities to continue browsing. The average time someone spends during a session on web pages and the total time spent at the site during the entire period of observation are among other time-related measures. Other variants on these variables can also be computed; for example Van den Poel and Buckinx (2005) used a variable named *Hurry* to indicate whether the average time of the clicks during the last session was less than the average over the past.

2.2 Navigation Metrics and Reports

Clickstream data collected automatically by application servers is the primary source of data representing the navigational behaviour of visitors. Depending on the goals of the analysis, this data can be transformed and aggregated at different levels of abstraction to provide metrics. In this section we review some fundamental metrics often used in the web usage context.

2.2.1 Website traffic

An important indicator in the web usage context is the traffic of the website, defined by the amount of data sent and received by visitors to a website (Kaushik, 2010). The traffic would be a measure to show how popular a website is. This can also be monitored on

individual pages or sections within a website. The following types of information are often used when monitoring web traffic:

Number of visits and unique visitors: The number of sessions that the website is visited over a specific time period, known as *number of visits*, is usually used as a web traffic indicator. Using anonymous unique identifiers in cookies we are able to identify if a visit is made by the same visitor (same machine) to the website during a specified time period. The number of visitors is another measure which shows the magnitude of the traffic on a website, sometimes referred to as *unique visitors*. These statistics are usually tracked and reported on a weekly or monthly basis to indicate whether there is any pattern of change. It is also possible to take advantage of control charts to differentiate whether a significant change from the natural variability of the metric occurs, based on a reasonable choice of time period unit.

Total number of pages visited: The total number of actions, usually the number of pages visited, carried out by users over a specific time period, can also be used to indicate website traffic. The total number of times which a specific web page is visited can be exploited to see if any change in the page, for example changing the design or content, results in more visits.

2.2.2 Website stickiness/slipperiness

In addition to the website traffic, which is the matter of attracting people to the website, it is important to encourage them to spend some time on the website and keep them interested in it. The web owner also wants visitors to re-visit the website. This concept is usually referred to as *stickiness* in the web analysis context, or sometimes known as level of *engagement*.

The stickiness of an e-commerce website is also related to the profit of the website, as a web owner wants visitors to see what the business can offer and how it can help them. The more interested visitors the website receives, the more likelihood they will buy from it. Stickiness can be calculated for a specified period using the average amount of time visitors spend on the website per visit (ATPV):

This measure can be calculated for the entire website, a section of the website, or even individual pages. If the web page has a link to a file, intended to be downloaded by visitors, the number of times the file gets downloaded can serve as an indicator of the stickiness of the web page.

Table 2.1: *Website traffic indicators for the specific time period of one week*

Variable		**Frequency**	**Percentage**
No. of visits	2	1652	15.7%
	3	193	1.8%
	4	77	0.7%
	5	28	0.3%
	+6	29	0.3%
Repeat Visits		1979	18.8%
Single Visits		8517	81.2%
Total		10496	

It should be noted that it is not always an advantage to have a high value of stickiness. When users are on *content pages*, which are designed to represent the products or information about the products, it is desirable for them to spend more time or have high stickiness. Adversely, for a *navigation* web page which is used to guide visitors somewhere else in the website, the more time spent, more clicks, or more pages visited may indicate that visitors are not being efficiently guided to the right section of the website. In this case a low value of stickiness is desirable. A low level of stickiness is sometimes referred to as *slipperiness*. In addition to the navigation pages, the high value of slipperiness for pages such as registration, shipment information, and transaction web pages indicates that they perform their functions appropriately.

Repeat visits versus First-time visits: The number of users who come back to the website after their first visit can provide important information about the websites stickiness. The high percentage of the sessions in which users have already visited the website at least once, referred to as *repeat visits*, can indicate how sticky the website is. The distribution of multiple visits/sessions also reveals how many people re-visit the website. For example, the average number of multiple sessions can serve as another metric to represent the stickiness of the website.

Table 2.1 reports the frequency of repeat and single visits. A large proportion of visitors, 72.2%, visited the website for only one session in the period of study; and there were 27.8% visitors of multiple sessions.

Bounce rate: The *bounce rate* refers to the percentage of visitors who come to the website but do not engage but rather leave the website after a few seconds or only visit a single page on the website rather than continue viewing other pages within the same

website. There is no complete agreement about the definition of bouncing and it may vary from site to site. Bounce may be defined as visiting only one page on the website, or it may be defined as single-page visits as well as visits which last five or ten seconds. The bounce rate can be calculated by

$$\text{Bounce rate } = \frac{\text{total number of bounced visits}}{\text{Total number of visits}} \tag{2.1}$$

The bounce rate can also be calculated for an individual web page. In this case, it is necessary to fix a time-out for the web page, which may vary depending on the content or navigation page. Then, the *page bounce rate* is the number of visitors who enter the site at a page and leave within the specified time-out period without visiting another page, divided by the total number of visitors who entered the site at that page.

2.2.3 Recency and frequency

The information about how often visitors return (frequency) and how long it is since they were last on the website (recency) are key metrics in customer behaviour study. Frequency and recency has been employed for many decades in business to identify active customers and achieve higher response rates to customer retention and loyalty efforts (Bucklin and Van den Poel, 2003). These measures have been used in business to develop a simple customer retention program (Glady et al., 2009; Fader et al., 2005). In the case of registered users, the web manager can identify a group of users for whom these measures exceed the limit. Therefore, it enables the web manager to implement a business act in order to influence customers' behaviour (for example by offering a discount) to encourage the customer to continue interacting with the e-business.

Frequency of visits: Frequency of visits by a visitor can be measured by the number of sessions a user visits a website. This might be limited to the period of time (e.g. the frequency of visits in a day/week). A related measure can help to distinguish between first-visit and repeat-visit to a website. This is important in the web usage analysis context, as it would help to know whether a visitor has previous knowledge about the website map in the current session. It should be noted that the average of the frequency measure of sessions may also provide a metric for the website stickiness.

Recency of visit: Recency in this context means the number of hours/days/weeks elapsed since the last visit to a website by a user. It is mostly a measure of web session quality. One popular measure is to use the average time between the last three visits (Glady et al., 2009), where it is referred to as *latency*.

Depth of visit: Besides the frequency of visits, another factor in web browsing behaviour is *depth of visit*. Depth of visit is a measure of visit quality showing how much a visitor interacts with the website in a web session. The more action by users on a website, the more successful the website to keep users in. However, the majority of visits to most websites worldwide contain fewer than 10 actions (Markov and Larose, 2007). Depth of visit is mostly used as an attribute for a web session. There are two variables in a web session which can be used for this purpose: the number of pages visited and the time the user spends on the website (Kaushik, 2007).

Number of pages visited: The number of pages visited (NPV) in a web session is the most popular measure to indicate depth of visit as a metric for the session. In the case of a repeat visit, another metric for a visit (or quality) is the total number of pages (or products) viewed during several sessions in a specific period of study. On the other hand, the average number of pages visited per session (ANPV) can present a measure for depth of engagement for a website. This is the fraction of the total number of pages visited (TNPV) over the total number of visits (TNV).

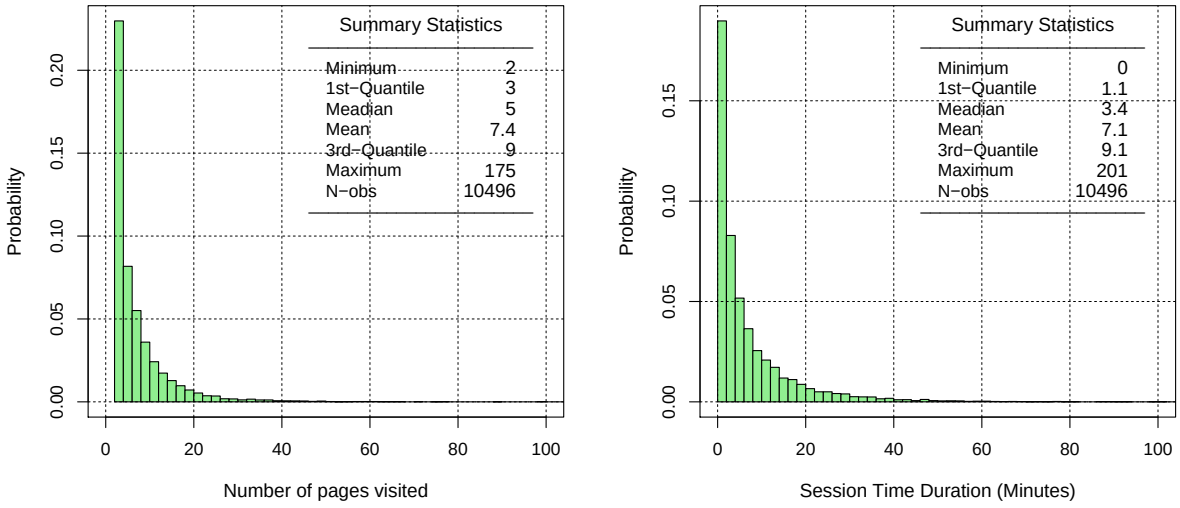
$$\text{ANPV} = \frac{\text{TNPV}}{\text{TNV}} \,. \tag{2.2}$$

It is also helpful to look at a histogram of NPV as a graphical display of tabulated frequencies of the number of pages visited. The histogram shows the entire distribution of depth of visits instead of a single statistic such as average number of pages visited per session.

Figure 3.1 (left) depicts the histogram of the number of pages viewed per session by visitors during the period of study, excluding the single-page visits. It also includes the summary statistics. The average number of pages visited was almost 7.5 pages for the period of study. The median indicates that half the users of the website viewed five or fewer web pages in a session. The minimum value is two pages, as we excluded single-page visits. The value for the upper quartile shows that 75% of sessions resulted in visiting less than 10 pages. A large number of sessions consist of only visiting two pages. Having many two-page visits is not good news for a web developer, as it might be due to a large number of users finding the website either not user-friendly or not relevant to their needs. However, it is not still disappointing, since most websites worldwide see the majority of users visiting fewer than 10 pages (Markov and Larose, 2007).

Session time duration: Session time duration is the amount of time a user spends on a website during a session. Logically, if users find the website interesting or relevant to their needs they are expected to stay longer. However, a high level of time spent on the website may not be considered as an advantage for an e-commerce website. For example,

Figure 2.1: *Histogram of the number of pages visited (left) and session visit time duration (right) for the period of study. Note that the single-page visits are removed*



an inappropriate web design may cause a customer to spend more time on the website to find the required item, to obtain the relevant information, fill in the shipment and payment information and generally take longer to complete online shopping than necessary. As a general rule, a good policy is to have a target goal which makes sense for the website.

As another metric, the aggregated version of time spent on site may be used. This way the web metric is the total amount of time a visitor spent on the website during a longer period (such as a week or month). One might decide to just consider the time duration for the last session, or the last K sessions before the current session. It is also possible to use a moving average of length K instead of total amount of time spent.

Similar to the number of pages visited, statistical summaries, such as average, median, or any other required statistic of the session time duration over several visits can provide a metric for the website depth of visit. Figure 3.1 (right) shows the histogram of the session time duration. To increase the granularity, the upper tail has been clipped at 100 minutes for this graph. The average session time duration is around seven minutes. Since the session time duration is a skewed distributed measure, we may need to look at the median as a summary statistic to represent the central tendency. The median session duration is about 3.4 minutes, which seems to be a more realistic estimate of the typical session time duration for those who visited more than one page. It implies that half of the sessions last less than 3 minutes and 20 seconds.

Average time duration per page: The average time users spend per page over all successive web pages viewed can represent the depth of visits to a website using both session time duration and number of pages visited. It can be computed by session time duration divided by number of pages viewed.

$$\text{ATD} = \frac{\text{TD}}{\text{NPV}-1}\,, \tag{2.3}$$

where TD is the total time spent on the website and NPV represents the total number of pages visited. The number of pages visited needs to be subtracted by one unit at the denominator, as the last page visited is not counted as a part of the visit time duration, but in the number of pages visited. Average time duration per page may be computed only for the time spent on the content pages, excluding the traversal pages to give a more reasonable metric showing depth of visit.

Table 2.2: *Table of metrics of the website using the SLC data set*

Measure	value	Measure	value
Number of visits	17400	Average time per visit (Min.)	7.1
Number of unique visitors	15789	Average time per page (Sec.)	63
Number of repeat visits	2890	Number of Pages visited per visit	7.4
Number of page visited	88904	Bounce Rate (%)	41.3
Total Time spent (hour)	1250	Conversion Rate (%)	7.9

The equivalent metric using time duration per page can be obtained for the website by the total time spent on it (TTD), divided be the total number of pages visited (TNPV)

$$\text{ATD} = \frac{\text{TTD}}{\text{TNPV} - \text{TNS}}, \tag{2.4}$$

where TNS denotes the total number of sessions, and ATD is the website average time duration per page. Because of the skewed distribution of the ATD, the use of median would be recommended, instead of arithmetic mean of time duration per page.

Hurriedness: In the web usage analysis, the analyst may want to distinguish short sessions on the pages from others. This can be produced by comparing the average time per page, extracted from all users behaviour on a website, with a specific level. For example, sessions whose average time duration per page is less than the average over all sessions can be grouped into the *hurry sessions*.

Table 2.2 shows some metrics of the website under study using clickstream data in SCL data set. It includes the traffic measures: Number of visits, unique visitors, and pages

visited. It also shows the total amount of time spent on the website by visitors which is 1250 hours. Stickiness of the website is reported through average time spent per visit which is nearly 7 minutes. The bounce rate also shows that 41% of visitors just visit a single page or leave the website within 10 seconds.

2.2.4 Conversion and Profitability

A successful conversion in internet marketing occurs when a visit is guided by the website to purchase a product(s) online. The conversion rate of a website, defined as the percentage of website visits that lead to an online purchase, is of great importance to e-commerce web marketers. Consequently, a great deal of effort has been made to improve the conversion rate by examining the motives of purchasers (Montgomery, 2001). It should be noted that conversion definition may vary for different types of website, and is mostly related to the aims of the website. In addition, a successful conversion can be defined for any type of desirable action from the web owner. For example, it may refer to actions such as a membership registration, newsletter subscription, software download, etc.

Number/Amount of conversion visit: The conversion can be measured by the number of visitors who make a purchase from the website as a percentage of the total number of visitors (TNV), referred to as *conversion rate*:

$$\text{Conversion Rate} = \frac{\text{Number of visitors who make an online purchase}}{\text{Total Number of visitors}} . \tag{2.5}$$

A typical conversion rate will be between 0.5%-8% depending on the sector, target market and quality of the website and proposition. Within online retail financial services, for example, 1 - 2% would be typical with 2% being very good (Kaushik, 2007).

Frequency of purchase: Frequency of purchase for the visitor can be measured by the total number of purchases a user ever makes at the site, usually available in the case of registered users. This measure may also be considered as a loyalty factor. For some analytical purposes it might be informative to compute the number of items purchased in the last visit to the website. Moreover, the analyst may be interested in using the average number of items purchased per visit.

Recency of purchase: The time elapsed between purchases has been proved to be an informative measure in off-line settings to find active and profitable customers versus non-active customers who shopped a long time ago (Wu and Chen, 2000). This measure has also been investigated and found effective for an e-commerce setting (Moe and Fader, 2004; Van den Poel and Buckinx, 2005). The recency of purchase can be measured by the

number of days elapsed since the last session which resulted in purchase. The summary statistics of this attribute over all visitors may serve as KPIs at the level of website.

Amount of purchase: The amount of money a user spends on the website is one of the most important factors to compute the *customer value*. There are several ways to measure the profitability of a visitor. For example, their total spending ever at the website, the average they spend on the website per session, or their average spending per session in which a purchase is made, all serve as measures related to visitor profitability. The analyst may only use the measures of amount spent during the last visit, or the moving average of amount spent at the last K visits.

Table 2.3 displays the different profitability measures computed for the SLC data set over the period of study. The conversion rate is reported as 13%, excluding bounce visits. This is a high conversion percentage as most websites have a conversion rate of less than 5%.

Table 2.3: *Table of conversion statistics for the SLC data set*

Measure	**Value**
Number of conversion visits (#)	1354
Total amount of conversion (£)	51,594
Average number of pages to purchase (#)	9.6
Average conversion time (Min.)	15.0
Time duration per conversion (Min.)	55.2
Conversion rate (%)	7.4

Staying on a website without making any purchase is not an advantage for an e-commerce website, as it may result in slowing down the server (similar to the customers in a restaurant or coffee shop). For an e-commerce website the time duration can be adjusted by the number of online shoppers. The time duration per conversion (TDPC) shows what amount of time spent on website by visitors provides a conversion for the website.

$$\text{TDPC} = \frac{\text{Total time duration}}{\text{Total No. of conversion}} . \tag{2.6}$$

However, for an e-commerce website that sells complex products/services, a longer visit duration may indicate a visitors interest in obtaining information about the products/services, or it may simply reduce the telephone service time, as customers get information from the website.

Considering the amount of purchase, the analyst can investigate the performance of the

website in terms of the time duration per hour/minute/second (ACPH).

$$\text{ACPH} = \frac{\text{Total amount of conversion}}{\text{Total time duration}}\,, \tag{2.7}$$

The ACPH shows what amount of revenue the website makes per hour/minute/second. Instead of time spent, it is also possible to look at the number of pages viewed by shoppers on the website before buying the first item. A related metric to represent the e-commerce website performance is called first purchase momentum (FPM) given by

$$\text{FPM} = \frac{\text{Required click to first purchase}}{\text{Average of actual clicks to first purchase}}\,. \tag{2.8}$$

The required click to first purchase needs the knowledge about the topology of the website. That is, if a visitor who aim to buy an product through the website could complete the order by visiting K page (or performing K actions), where K is the minimum number of action for this purpose. The average number of clicks made by visitors who buy that item from website is calculated using clickstream data. Hence, the FPM assumes values between 0 and 1. A FPM close to 1 shows the clarity of the content and navigation on the website.

Average conversion time: It is useful to provide a metric using the ratio of conversion time and session time, since this can be used to investigate at what point in a session a user decides to make an order. A desirable metric for an e-commerce website is given by the average amount of time shoppers spend on the website to buy an item online. A shorter conversion time shows a good performance by the website in terms of guiding visitors for e-shopping.

Table 2.3 shows some conversion statistics for the website using SLC data set, including the total number of sessions in which an online purchase occurs and the total amount of conversion. The average number of pages visited to the first purchase, in the group of conversion sessions, is reported as 9.6 pages and on average it takes place around 15 minutes after the session starts. Time duration per conversion shows that for every 55 minutes that visitors surf the website one conversion occurs.

2.3 Trend and Segmentation Reports

Several important web analytic reports can be produced by drilling-down the web metrics into segments of different web session attributes. The resulting report can either be utilised to compare several segments with respect to the metric, or it gives the measures for the specific subset of the visit. For example, depth of visit metric may be calculated for

visitors who made an online purchase and those who finished a session without purchasing. As another example, the traffic reports of a website can be reported only for a specific geographic location. Finding the website metrics for natural ordering of time would also provide the ability to observe potential trends in metrics. This would be achieved by computing the measures for different time periods such as day, week, or month. In this section we review some reports available in the clickstream data by which more knowledge can be obtained about aspects of website or web browsing behaviour through trend and segmentation reports.

2.3.1 Segmentation Reports

Segmentation of web metrics aims to provide a better perception of how the key factors would affect the metrics. It might be implemented based on certain defined actions in the website, such as adding an item to the shopping cart, clicking on an online advertisement, completing a registration form, or most importantly completing a payment transaction.

Geographic location: Web server log files provide information about the internet service provider (ISP), the company that offers its customers access to the internet. This naturally gives us some information about the area from which a user of the website connects to the world wide web. This information can be used, for example, to report on the number of users who connect from the domestic area (UK in the SLC data), or if the connection has been made through an ISP outside the country. The analyst may be interested in tracking whether there are some specific web pages that are being viewed mostly by people from a specific area or country.

Table 2.4: *Segmentation report of the sessions in which users come to the website by means of UK internet service providers or other non-UK ones*

Variable	PNV	BR	NPV	TD	CPV	CR
Non-UK Visitors	48.6	46.8	4.1	51.3	1.2	3.1
UK Visitors	51.4	36.3	5.2	53.4	6.3	14.6

PNV: Proportion of the number of visits (%)
BR: Bounce rate (%)
NPV: Number of pages viewed per visit (#)
TD: Average time session duration (Seconds)
CPV: Amount of conversion per visit (£)
CR: Conversion rate (%)

Table 2.4 shows that about 51.4% of the visitors to the website are located in the UK.

It seems that the bounce rate is higher for the non-UK visitors. The depth of visit for visitors who stay on the website, excluding bouncing visitors, seems to be slightly larger for UK visits than non-UK ones. However, there is a considerable difference between conversion attribute, as the UK visitors spend 5 times more than non-UK visitors on the website. The conversion rate is also reported to be much higher for the UK visits compared to non-UK visits.

Referring website: Users arrive at a website either by typing the website address or by clicking on a link given by intermediate websites, such as advertisement links or search engines. A common way of finding a website link is by searching via motor search engines such as Google, Yahoo or Bing. Each website has a proportion of its traffic produced by search engines. An important report in web analytics is the traffic of the website received from different referring websites, the website from which the visitor comes to the website. Reporting the ordered conversion rate for different referring websites can help to show those which send the target population to the website.

Searching keywords: Many web analytic services provide reports that show the search terms used by visitors who found the site through search engines. This type of report can give knowledge about advertiser bidding and help to optimise the bidding over search keywords of the website. For more information see Kaushik (2007).

First-time/Repeat session: The website metrics can be broken down based on whether the session is a first-time session or a repeat one. For example an analyst may be interested in calculating the average time spent on the website for these groups. More importantly, they can find out the percentages of conversion for first-time visits compared to repeat visits.

Technical machine information: Clickstream data contains information about the types of browsers and operating systems used. This information may be useful to web designers for optimising the website for the appropriate technical capabilities that help visitors use the website.

Visitor demographics: Demographic information of visitors gives some information about the population visiting the website. This kind of information is only available for registered users of the website. The demographics may include: gender, age, education level, household size, number of children, income, language spoken, race, or other relevant information collected at the registration step.

A visitor becomes a registered user of the website by providing some credentials, usually in the form of a username (or email address) and password. This way, a visitor can access

information that is unavailable to a non-registered visitor. Non-registered visitors are usually referred to as guests. A clickstream of registered users can be identified when they provide the credentials for a website which is usually called logging in, or signing in.

Registration may also help to produce other measures. For example, a measure of trust can be defined by whether a visitor did or did not put his telephone number or profession at the site owner's disposal when registering (Van den Poel and Buckinx, 2005; Li et al., 2002). Drilling down the metrics based on demographic information can help to reveal some patterns if it affects the web browsing behaviour of visitors. For example, users who speak a different language to the language of the website may be found to have longer session time duration.

2.3.2 Trend Reports

The web analyst may drill down the metrics such as the number of visits, the number of pages visited, session time duration, by time/date attributes of the visit made by visitors to a website, which are known as *trend reports*. These reports can provide the analyst with a better perception of any required metric as they show the changes of a metric over a time period, in addition to the metric for the whole period.

The time-stamp available in the clickstream helps to find the *visit-date*, the date of a new web session in which the first request of the user's browser is sent to the server. Visit-date information enables us to derive other temporal session attributes such as whether the session takes place on weekdays, holidays, weekends, or any required period of interest. It should be noted that weekends and holidays in different countries are not the same, so it is necessary to take into consideration the country in which the user is located when browsing the website.

The same report can be produced based on *visit-time*, the time spent by the server on a company's website. As visitors may connect from different time zones it is necessary to know the local time at which the visitor connects to the website. In order to compute the local time, remote IP longitude available in clickstream data is used to indicate how far ahead or behind Greenwich Mean Time (GMT) the local server time is. Every other variable related to the time of the visit can be extracted from this field. Depending on the purpose of the analysis, any other alternative time category might be used.

It is usually desirable to find the website traffic for different hours in a day in order to find the most popular viewing time of the site. This would show the peak time of the traffic, may be used to find a suitable time to do promotional campaigns. On the other

Figure 2.2: *Line chart shows the trend of the number of visits (left) and the percentage of visitors referred from Google ads to the website (right) over a period of one week.*

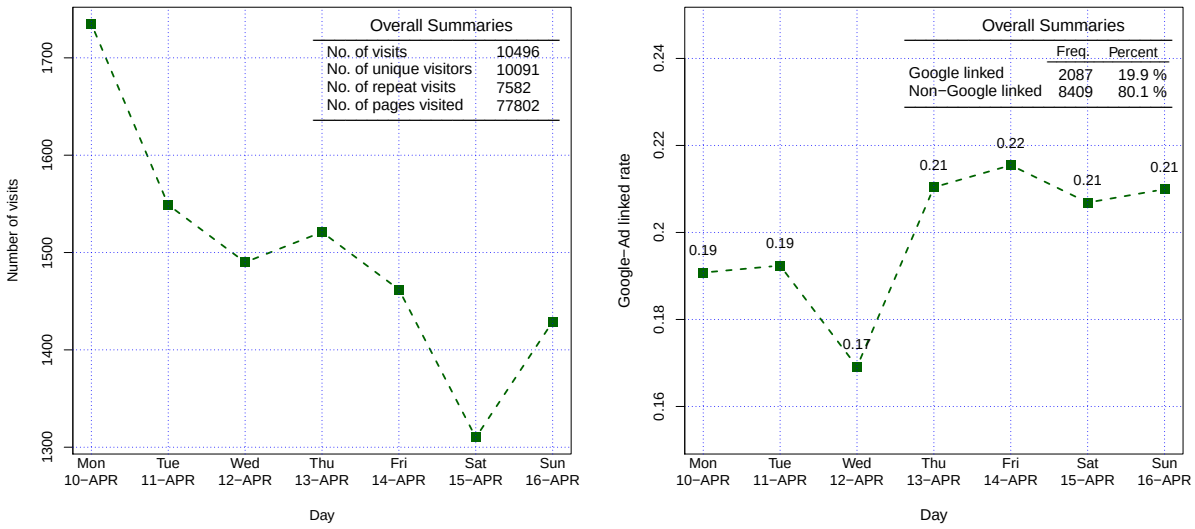
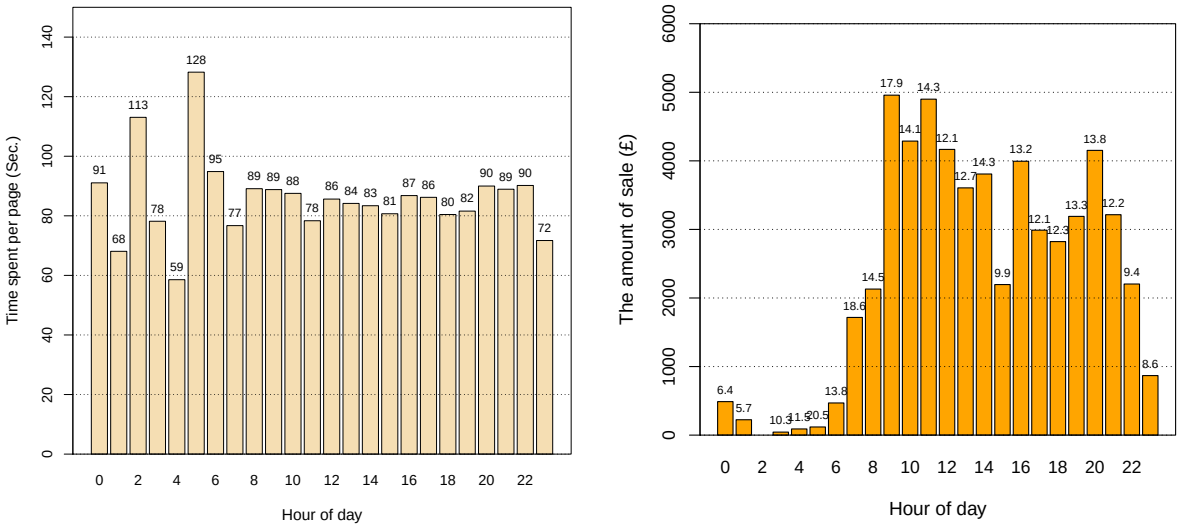


Figure 2.3: *Bar-chart for the average time spent per page (left) and the amount spent on sale for different hours of a day, where the conversion rate for each level is represented at top of the bar (right).*



hand, the websites low traffic hours would be ideal for performing maintenance activities.

Figure 2.2 (left) shows how the number of visits changes over a one-week period. It seems that the website traffic has a decreasing pattern during a week. It also contains the web traffic indicators for the whole period of study. Figure 2.2 (right) depicts the trend in the percentage of visits referred to the website from Google during a one-week period. It also shows that, in total, 20.0% of the websites traffic is from Google.

Figure 2.3 (left) shows the average time spent per page in different time of the day. In order to exclude the different hours, the graph only shows the statistics for domestic visitors connected to the website from the UK. Ignoring the visits from 1:00am–5:00am, it seems that the average time spent per page is nearly the same at different times of the day. Figure 2.2 (right) shows the amount of sale for the website in different times of the day. As would be expected, the highest amount of sales takes place during day time, from 9:00am–12:00pm. The width of the bars is proportional to the conversion rate for the corresponding time segment.

2.4 Web Page and Traversal Reports

As well as the general information about a website and sessions such as traffic, depth, time/date, time duration, depth of a session, etc., clickstream data provides information about the sequence of web pages a user visits while browsing the website. The sequences of web pages visited in a session can be analysed to determine which path would result in a desired outcome, for example an online purchase, visiting a certain part of site, etc. One primary web traversal analysis is to investigate the entry/landing pages, the page at which a user starts his/her session, and exit pages from a session. For example, a web manager might be interested in finding the landing pages which are more likely to guide users to a specific part of the website. It is also informative to be able to measure the amount of time people spend on different pages or sections of the website. In this section we review some metrics and reports based on the traversal path visitors take on the website.

Top entry and exit pages: Clickstream data provides information about the most-used entry pages, the page by which users get onto the website in a session is usually referred to as an *entry/landing* page of the session. Regarding entry pages, the percentage of visitors who arrive at the homepage is highly important as website designers aim to design the homepage in such a way as to give the best impression to visitors. It would also be informative to drill down the stickiness for different top entry pages. For example, the

bounce rate or stickiness for each of the top entry pages might be produced to see if they have a high bounce rate.

The analyst can report the top exit pages, the last pages visited just before the user left the website. This might give the web designer an indication as to why visitors leave the website. If this leakage point can be determined, the websites performance could be improved. The exit pages can also reveal navigation characteristics about the website. A related metric can be found based on the proportion of users who leave the website from a specific page, or part of the website, sometimes known as the emigration rate for the page:

$$\text{Emigration rate} = \frac{\text{Average number of exit from the page/section}}{\text{Average number of visit to the page/section}}. \tag{2.9}$$

Table 2.5 shows the first five top-entry pages and the trends of the number of visits over a one-week period. This can be plotted afterwards by a trend plot with 5 colour lines. It is reported that only 17% of the visitors to our SLC data set arrived at the homepage. The first top entry page are product pages, 52%.

Top error rate pages The error rate for the web page or section is calculated by the total number of times a page goes to the error pages

$$\text{Error rate} = \frac{\text{The number of errors appearing for the page/section}}{\text{the number of visits to the page/section}}. \tag{2.10}$$

The most popular paths: A primary report finds the most frequent paths visitors take on the website. From a web design point of view this report is highly important as it shows whether the visitors follow the path that the web designer wanted them to follow. If not, it gives some indication as to how to redesign the website structure. It should be noted that on most websites the most popular paths are usually followed by a small percentage of visitors, usually 1%. Thus, it is questionable whether one can make any decision with such a small fraction of site traffic (Kaushik, 2007).

Another report can show the traversal between page/section to page/section, where a higher percentage of traversal is denoted by a thicker arrow between pages. For example the table of the most popular paths through the website, from the homepage to the contact page.

Table 2.5: *The most popular landing pages of the website.*

Rank	Landing page	Frequency	Percent (%)
1	/shopping.php	3943	52
2	/home.php	1289	17
3	/choosingshaver.php	607	8
4	/search.php	379	5
5	/philishave.php	303	4

2.5 Discussion

In this chapter, we discussed how the measurement, collection, and reporting of clickstream data can provide valuable information to enable us to understand website usage and performance. In addition to the common application of clickstream for website traffic, engagement and depth we used the conversion information to report some KPI regarding the profitability of the website. As well as the metrics produced for the website, we reviewed the metrics that can be extracted from clickstreams, showing attributes of the web sessions. These measures will be used for different analytical purposes throughout the thesis.

Chapter 3

Analysis of Clickstreams: Depth of Visit

In this chapter we aim to probe the SLC data in more depth using summary statistics, graphs, tables, and basic statistical tests. The focus is on the number of pages visited (NPV) and session time duration (TD) as metrics which indicate the depth of a visit. We will investigate these variables individually and also try to reveal the inter-relationship between measures. As well as some basic explanatory analysis of web session data, we address some problems regarding the use of standard statistical tools such as the two-sample t-test, and a goodness-of-fit test for large sample size clickstream data. We illustrate how graphical tools equipped with effect size measures can give the analyst an alternative in order to assess practical significance along with statistical significance.

3.1 Introduction

Analysis of the clickstream data offers the opportunity to enhance an understanding of a website and prediction of the website's visitor behaviour (Andersen et al., 2000; Markov and Larose, 2007). The statistical analysis of session data constitutes the most common form of analysis. The resulting knowledge can be potentially useful for improving the website's performance and, specifically, to provide support for marketing decisions for an e-commerce website.

Despite the presence of difficulties in using clickstream data, a great deal of effort has been put into the study of browsing behaviour using clickstream data. A descriptive study of clickstream data can be found in Catledge and Pitkow (1995). Bucklin and Sismeiro

(2003) developed a model for analysing internet browsing in terms of the probability of leaving the website or selecting another page. They also modelled visiting time duration for a single web page using covariates such as the number of bytes transferred, cumulative number of pages viewed prior to the current page, a reload request for a page, having dynamic content, advertising on a web page, an error occurring in a page transfer, and server response time. Danaher et al. (2006) applied a random effects model to determine the impact of some factors on session time duration and the number of pages viewed. The longer the page visit lasts, the more likely it is to keep the visitor interested in the site and consequently gives users more time for conversion (Hanson, 2000). Moe and Fader (2004) showed that repeat visits and greater long-term sales are associated with user interest. Demers and Lev (2001) illustrated that websites with longer visit session duration show higher monthly profit. Hence, visit time duration may serve as an indicator of the future earnings of a website. Chatterjee et al. (2003) investigated the association between repeat-visits and conversion rate. The depth of search has been investigated by Johnson et al. (2004), using an exponential gamma timing process. Moe and Fader (2004) studied the loyalty of users over time based on the frequency of visits to specific websites and showed that frequent shoppers are more likely to purchase in subsequent sessions. Park and Fader (2004) made an inference about individual browsing behaviour using multivariate timing mixture model for modelling cross-site timing behaviour. Bucklin and Sismeiro (2004) reviewed major developments from the analysis of clickstream data, covering advances in e-commerce context. It also discussed the inherent limitations of clickstream data for understanding and predicting browsing behaviour.

3.2 Distribution of Number of Pages Visited

In this section we describe SLC clickstream data by examining and summarizing the distribution of each individual variable. We take advantage of tables of frequency, plots and summary statistics to obtain basic knowledge about variables studied. Summary statistics provide information about central tendency, dispersion, and the shape of the overall distribution. Nevertheless, researchers are mostly interested in finding a mathematical model which explains the probability behaviour of the variable. Having found a mathematical model, the analyst would be able to answer any probabilistic question about the variable under scrutiny, as well as all measures of tendency, dispersion, shape, etc.

A key factor of customer behaviour on a website is the number of actions (usually the number of pages visited) made by visitors during a session on a website. From a web developing point of view, when visitors to a website perform more actions it can be

interpreted as the website having efficiently engaged the visitor. However, the majority of visits to most websites worldwide contain fewer than 10 actions (Markov and Larose, 2007). In general, it is very important for web managers to detect if visitors leave a website quickly. We are interested in finding a statistical distribution to explain the probability behaviour of the number of pages visited. This way, we can infer about the probability that a user leaves a website after visiting n pages. It can also establish a way to compare websites with respect to the number of actions visitors perform in different time periods to recognize when a considerable change from previous patterns takes place (see box 3.1).

Questions 3.1 How many pages do people view (or load) during a visit session?

Hypothesis 3.1 The probability behaviour of the variable NPV can be explained by Weibull distribution, NPV $\sim$ Weibull(α, λ)

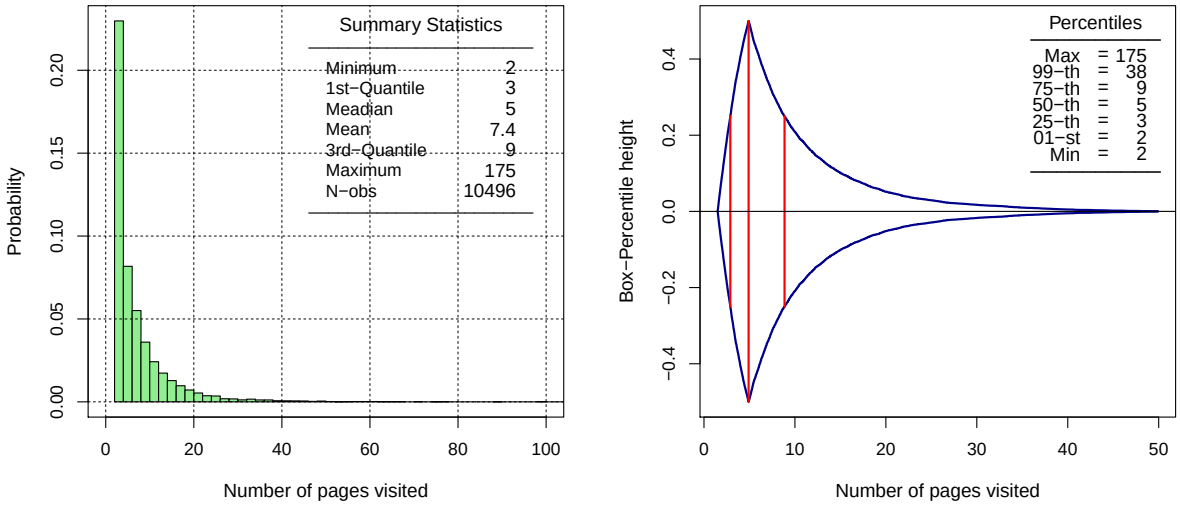
Approaches 3.1 The test to investigate whether the sample of data is consistent with a specified distribution function (e.g. Kolmogorov-Smirnov, Chi-Squared goodness of fit test), QQ-plot visually helps to probe whether data follows a reference distribution

3.2.1 Graphical representation

A primary analysis is to use summary statistics to obtain a perception of quantity of visit. It is also helpful to look at the histogram of number of pages visited as a graphical display of tabulated frequencies of the variable. Plots and numerical summaries play a crucial role in statistical analysis. Particularly, we take advantage of plots to present the data before entering a modelling step, and also as diagnostic tools after modelling the data (Chambers and Hastie, 1993). Figure 3.1 displays the histogram of the number of pages viewed for users of the website. We showed this histogram as a measure of depth of visit, in chapter 2. The histogram emphasize that NPV has a very right-skewed distribution.

The boxplot is a common way of representing numerical data through its five-number summaries: minimum, lower quartile, median, upper quartile, and maximum (Chambers et al., 1983). It also serves as a quick way of representing the distribution of one or more quantities graphically. It may seem more primitive than a histogram or kernel density estimate but it does have some advantages. Specifically, it takes less space and is potentially useful for comparing the distribution of a quantity over several groups or several quantities. On the other hand, contrary to the histogram, the appearance of which can be heavily affected by the choice of number and width of bins, the boxplot, to some extent,

Figure 3.1: *Histogram (left) and box-percentile plot (right) of the number pages visited for multiple-page visits sessions.*



is a robust tool for representing a distribution. There is a similar problem for a kernel density estimate, as the choice of band-width can heavily influence its appearance. A boxplot is also used to indicate which observations potentially can be considered outliers.

There is a difficulty in using the standard boxplot for a large number of observations, as it displays a lot of data records as outliers. This problem becomes more critical when data have a skewed distribution. In practice, not all values found beyond the whiskers in a boxplot are outliers, but this is the result of the expectation that distributions of data should be Normal. The outer box of the boxplot can be increased by increasing the coefficient of the inter-quartile range (IQR) from 1.5 to a larger value. For example, a boxplot with 3×IQR covers a larger proportion of the data points with its whiskers. However, a large number of data points may be located outside the whiskers for long-tailed distribution. It also does not make a good picture of the tail of the distribution.

Hubert and Vandervieren (2008) introduced an adjusted boxplot using the robust measure of skewness, *medcouple*. Taking into account the skewness, the adjusted boxplot try to reasonably cover tails of the distribution. This approach also helps not to flag many regular observation as potential outliers. The function `adjbox` in `R` software, available as part of the `robustbase` package, plots the adjusted boxplot based on Hubert and Vandervieren (2008). This function did not give us the adjusted boxplot for the NPV variable. The problem is that the available algorithm for computing medcouple fails for

large data sets (Brys et al., 2006). Meanwhile, for large sample size observations the adjusted boxplot may represent many overlaid points located outside the whiskers. That is, it does not provide a suitable graphical view of the tails.

An alternative way is using a modified version of the standard boxplot which represents the 1st, 5th, 12.5th, 25th, 50th, 75th, 85.5th, 95th and 99th quantiles on the graph. At any height the width of the irregular box is proportional to the quantile of that height, up to the 50th percentile, and above the 50th percentile the width is proportional to 100 minus the quantile. Thus, the width at any given height is proportional to the percent of observations that are more extreme in that direction. This way, the modified boxplot represents data by boxes of different width which contain intervals of 0.99, 0.90, 0.75, 0.50 proportion of data. This approach can be extended to produce a boxplot by considering all percentiles of the data. The result is known as box-percentile plot, providing more information about the univariate distributions. The median, lower quantile and upper quantile are marked with line segments across the box (Esty and Banfield, 1992).

The large number of data records in our study and the long-tailed distribution of NPV caused us to choose the box-percentile as an alternative to the ordinary boxplot to represent this variable. Figure 3.1 (right) shows a box-percentile plot of the number of pages viewed. We assign the height of 1 as the maximum height at 50-th percentile or median (plotted from -0.5 to 0.5). The height of the plot for at p-th percentile when $p < 50$ is $p/100$ and for $p > 50$ is $(100-p)/50$. For example at 25-th percentile (at x=3) the height is half of the height of graph at median, $25/50 = 0.5$. The same height represents the 75-th percentile, as it is located after median. The short distance between minimum value and the lower quartile, and also between lower quantile and the median, implies that the data values have been located in a relatively small range of data. On the other hand, the large gap between the upper quartile and 99-th percentile emphasises a long right-tailed distribution.

3.2.2 The Weibull distribution

The Weibull distribution is a continuous probability distribution used since 1951 to describe the statistical behaviour of phenomena (Weibull, 1951). The probability density function of a Weibull random variable X is:

$$f(x;\, \lambda, \alpha, \theta) = \frac{\alpha}{\lambda}\left(\frac{x-\theta}{\lambda}\right)^{\alpha-1} \exp\left[-\left(\frac{x-\theta}{\lambda}\right)^{\alpha}\right] \qquad x \geq \theta \tag{3.1}$$

where θ, $\alpha > 0$, $\lambda > 0$ are called the threshold, shape, and scale parameter respectively. In some literature, this distribution is referred to as the *shifted Weibull distribution* (Johnson

et al., 1994). In the case of $\theta = 0$, the pdf given in 3.1 reduces to the density function of the standard two-parameter Weibull distribution. The Weibull distribution is often used in the field of life data analysis, due to its flexibility, as it resembles the behaviour of other statistical distributions such as the Normal and the Exponential by various values for shape and scale parameters. If we consider stopping browsing a website as a failure, the NPV is a failure variable, as the session with NPV= k means that the visitor has left the website after the k-th page visit. The Weibull distribution is defined by shape, scale, and threshold parameters. The parameter α describes the shape of the Weibull curve. A shape of 3 approximates a normal curve. A shape between 2 and 4 is still fairly normal. A small value for α, say 1.25, gives a right-skewed curve, where large values, say 10, give a left-skewed curve. The scale parameter in the Weibull distribution is matched to the 63.2 percentile of the data. The scale defines the position of the Weibull curve relative to the threshold, analogous to the way the mean defines the position of a normal curve. A scale of 5, for example, says that 63.2% of the sessions will leave the website in the first 5 web pages following the threshold (two pages in our study). Threshold is a shift of the distribution away from 0. A negative threshold shifts the distribution to the left of 0, and a positive threshold shifts the distribution to the right of 0. All data must be greater than the threshold.

It should be noted that the number of pages visited is an integer quantity. Hence, when using the Weibull distribution in order to approximate the probability behaviour of this variable we need to implement correction of continuity to fix the discontinuity. For this reason, we add a random number, u, from uniform distribution between -0.5 to 0.5, $U \sim \mathcal{U}(-0.5, 0.5)$ to the original NPV observations. The NPVc = NPV $+u$ is used throughout this section.

3.2.3 Parameter estimation

There are often a number of different approaches available for estimating parameters of the distribution given a data set. Depending on the statistical properties of the method, an analyst may need to choose the appropriate method to apply, the desired application of the fitted model or even the relative difficulty in applying a method.

Graphical Method: A straightforward way to estimate the parameters of the Weibull distribution is given based on the following relationship:

$$\ln \ln(\frac{1}{1 - F(x - \theta)}) = \alpha \ln(x - \theta) - \alpha \ln(\lambda) \tag{3.2}$$

The relationship given in 3.2 provides a way of checking whether data follows the Weibull distribution. That is, the plot of logarithm of data versus $\ln\ln(1/[1-\widetilde{F}(x-\theta)])$, when data follows the Weibull distribution, is expected to be located on a straight line with a slope α and the intercept $\alpha\ln(\lambda)$, where $\widetilde{F}$ denotes the empirical cumulative distribution function of the data. Graphical methods are usually used because of their simplicity and graphical appeal (Abernethy et al., 1983).

In the case of having a straight line fitted, the slope and intercept can be used to estimate α and λ. As the parameters for linear regression are usually estimated through least squares, corresponding parameters of the Weibull distribution found this way are referred to as least square estimates in some literature. This plot is depicted in Figure 3.2 (left). Due to omitting the single-visit sessions from the SLC data, the number of pages viewed starts from 2. Therefore, it is reasonable to use a shift (or location) parameter of $\theta = 2$. Figure 3.2 (right) shows the QQ-plot of the NPVc, empirical quantiles of the NPVc versus quantiles of the shifted Weibull distribution, where the shape and scale parameters are $\widehat{\alpha} = 0.76$ and $\widehat{\lambda} = 4.84$ respectively. The plot shows that the theoretical distribution reasonably explains the distribution of NPV variable.

The QQ-plot is a common graphical technique for comparing two data sets, either two sets of empirical observations, or one empirical set against a theoretical set. It helps to assess how closely two data sets agree by plotting the quantiles of two distribution functions against each other. The 45-degree line gives a graphical measure of the resulting goodness of fit so that it shows the difference between the sample set and the theoretical distribution. It can also be used as a graphical method to find the regions or the range of values that theoretical distribution does not fit well, via departure from the 45-degree line (Gibbons and Chakraborti, 2003). As can be seen from Figure 3.2 (right) most of the points lie on the 45-degree line. There is a small departure from the straight line at the far end of the right tail (the values larger than 99-th percentile). Hence, the QQ-plot shows a reasonable fit of the theoretical distribution over the number of pages visited in the website, but it slightly deteriorates at the extreme end of the right tail. However, it is difficult to infer if this departure from the straight at tail is due to miss-specification, as QQ-plot for many sample experiment show an slight deterioration from straight line.

Figure 3.3 (left) depicts how well the fitted Weibull distribution curve matches on the histogram of the number of pages viewed. In the Figure 3.3 (right) the cumulative density function (cdf) of the fitted Weibull distribution is also very close to the empirical cdf of the NPV variable.

Maximum likelihood and Method of Moments: The classical Maximum Likelihood

Figure 3.2: *The logarithm of the NPV versus the* $\ln\ln(1-\widetilde{F})^{-1}$ *and the fitted simple linear regression (left). QQ-Plot of the number of pages visited (right).*

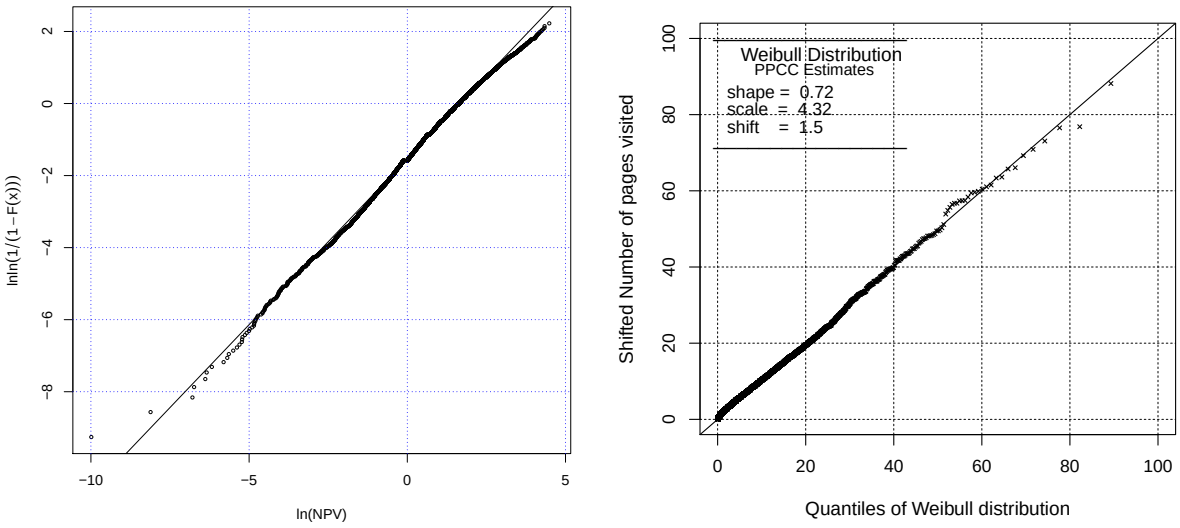
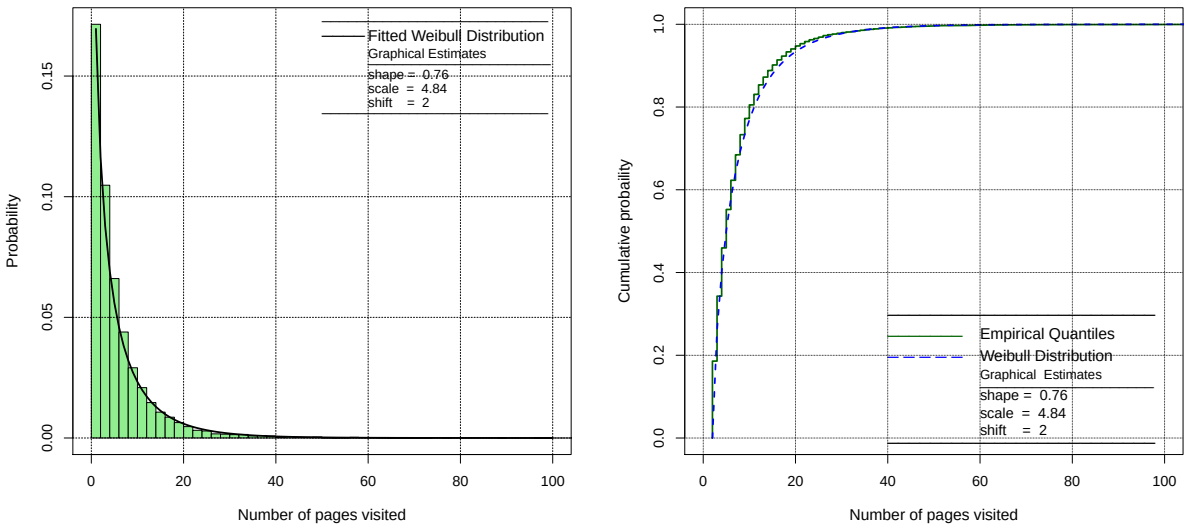


Figure 3.3: *The Weibull curve on the histogram of the number of pages visited (left) and the cumulative density function of the NPV and the fitted shifted Weibull distribution (right)*



(ML) approach can also be applied to estimate the parameters of the distribution. The ML method is mostly used because of its desirable analytical properties (Engelhardt, 1975). Another standard analytical method to estimate the model parameters is given by the *method of moments* (MM) in which the parameters are estimated through equalising the k-th theoretical and sample moments, for some values of k depending on the number of parameters of the reference distribution (Mann et al., 1974).

The ML estimators of the parameters of the Weibull distribution are not analytically tractable, so that numerical methods are used to estimate the parameters given in the data set. We used several different starting points to verify if the maximisation algorithm had reached the global maximum rather than the local, when finding the ML estimate using both `fitdistr` and `mle` functions. The ML estimates of the shape and scale parameters respectively are $\widehat{\alpha} = 0.87$ and $\widehat{\lambda} = 5.39$. Figure 3.4 (left) shows the QQ-plot of the NPV versus the theoretical model estimated by ML approach. The plot shows that the QQ-line deteriorates from the straight line for the right tail of the distribution, whilst it gives a reasonable fit for the range of values where there is a high percentage of observations. However, departure from the straight line at extreme right tale at the QQ-plot does not mean that the ML estimate is weaker than the other estimation methods. Figure 3.5 displays the QQ-plot to the point NPV= 20, the 95-th percentile of the number of pages visited. This way, we remove the tail so as to gain a better understanding of the GOF of the parameters estimation regardless of the right tail. It can be seen that the ML estimates show a closer fit for the range of values between (0 , 20) in comparison to the PPCC., when compared to PPCC.

Probability plot correlation coefficient (PPCC) method: The correlation coefficient between the paired sample quantiles, known as the probability correlation coefficient, is used as another technique to identify the shape parameter for a distributional family that best describes the data set, specifically for distributions with a single shape parameter (Filliben, 1975). It also serves as a measure of goodness of fit of a theoretical model to the observed data. The closer the correlation coefficient to one, the better the distribution fits the data. The correlation is represented on the graph known as *probability plot correlation coefficient* plot. The y-axis of PPCC plot represents the correlation coefficient between the empirical quantiles of the variable under study and the quantiles of the fitted theoretical distribution for different values of the shape parameter is given in x-axis. One can simply estimate the shape parameter by the value which gives the highest correlation coefficient. This technique is appropriate for family distributions that are defined by a single shape and scale parameter, like the Weibull distribution.

Figure 3.4: *The QQ-plot of the number of pages visited using ML estimate of the parameters (left) and using PPCC estimate of parameters (right).*

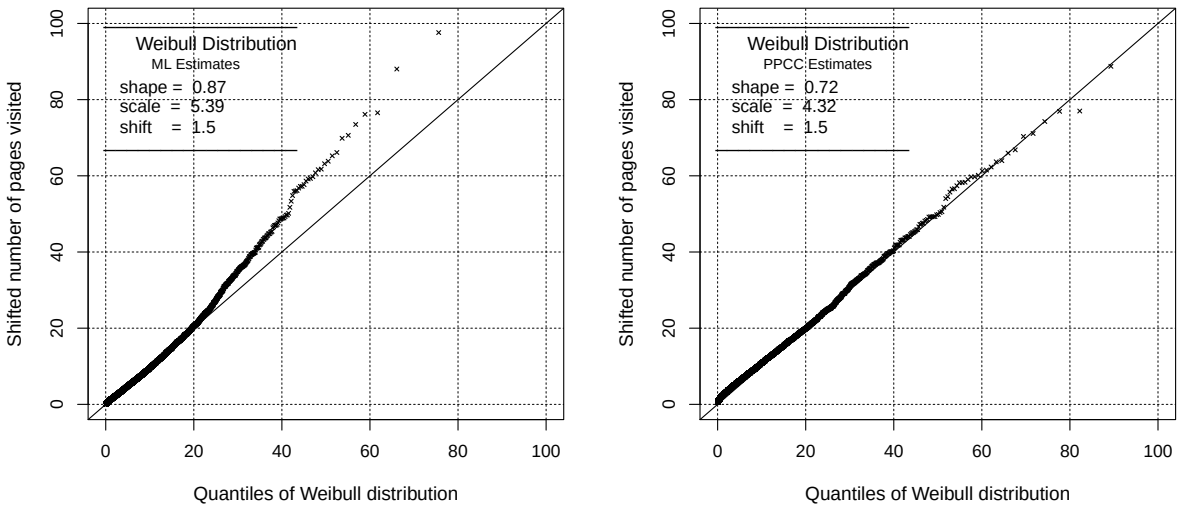


Figure 3.5: *The QQ-plot for the NPV versus the fitted Weibull distribution using ML estimates (left) and PPCC estimates (right), zoom in the range of values* $(0, 20)$

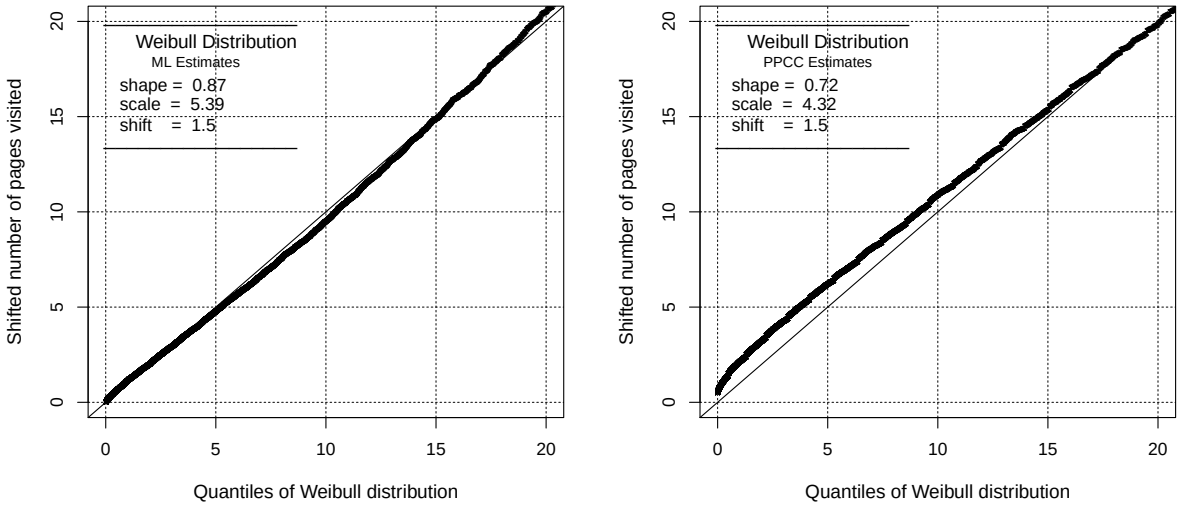
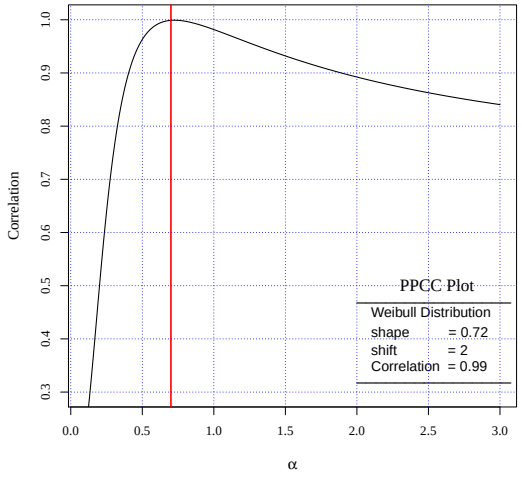


Figure 3.6: *The Probability Plot Correlation Coefficient (PPCC) Plot. The y-axis represents the correlation coefficient between the empirical quantiles of the NPV and the quantiles of the fitted Weibull distribution with corresponding shape parameter given in the x-axis.*



The PPCC method does not give any information about the scale parameter so the scale parameter is chosen after estimating the shape parameter by calibrating the theoretical quantiles to provide the better QQ-plot over the range of quantiles. This way, the scale parameter is estimated through a graphical tool. Figure 3.6 shows the PPCC plot for the NPV variable. The optimum value for the shape parameter $\widetilde{\alpha} = 0.72$ gives us an estimate for the shape parameter and the scale parameters $\widetilde{\lambda} = 4.32$ provides a reasonable QQ-plot over the range of possible distributions. Figure 3.4 (right) depicts the QQ-plot of the NPV versus the theoretical model estimated by PPCC approach. The graph shows that there is a good match between the empirical distribution and the theoretical fitted distribution in the corresponding QQ-plot and the fitted model reasonably explains the distribution of NPV.

3.2.4 Goodness-of-fit test

The *goodness of fit* (GOF) of a statistical model measures how well it describes a set of observations. Measures of goodness-of-fit typically summarise the discrepancy between observed values and the values expected under the model in question. Such measures can also be used in statistical hypothesis testing. For example, the Kolmogorov-Smirnov (K-S) statistic helps to test whether outcome frequencies follow a specified distribution.

As another example, Pearson's chi-square test can help to test whether two samples are drawn from identical distributions.

Kolmogorov-Smirnov test: The non-parametric one-sample K-S test helps to investigate whether data follows a theoretical probability distribution. It compares an empirical cumulative distribution of a sample with a reference probability distribution. The KS test can be applied only for continuous quantities, so we use the NPVc variable to test if the Weibull distribution provides a good fit. The test is performed under the null hypothesis that the NPVc follows the Weibull distribution with estimated shape and scale parameter. The K-S test statistic for ML estimates is $D = 0.1348$ and the corresponding p-value is less than 2.2×10^{-16}, which reject the hypothesis of the NPV being followed by Weibull distribution at any level of significance. The result is not surprising as, in the case of the large sample size, standard statistical tests tend to be significant even for small departures from the hypothesis. Table 3.1 shows the p-value of the KS test for all three estimation methods.

Chi-square test for goodness-of-fit: We also inspect whether the NPV variable comes from a population with a specific distribution by the chi-square test for goodness-of-fit. The chi-square goodness-of-fit test can be applied to any univariate distribution using the cumulative distribution function. The chi-square goodness-of-fit test is applied to binned data, where each observation is allocated into different classes. This is in fact not a restriction since frequency table can simply be calculated for non-binned data before implementing a chi-square test. However, the value of the chi-square test statistic depends on how the data is binned. This method requires a sufficient sample size in order to have a valid approximation of chi-square distribution. The chi-square goodness-of-fit test can be applied for both discrete and continuous distributions, unlike the K-S test which can be applied only for continuous distributions. The chi-square test is defined for the null hypothesis of H_0: The data follows a specified distribution; against the alternative hypothesis H_a: The data does not follow the specified distribution. For the chi-square goodness-of-fit computation, the data is divided into K bins and the test statistic is defined as:

$$\chi^2 = \sum_{i=1}^{K} \frac{(f_i^o - f_i^e)^2}{f_i^e} \tag{3.3}$$

where f_i^o and f_i^e are respectively the observed and expected frequency for the i-th bin. The expected frequency is calculated by $f_i^e = N \times [F(u_i) - F(l_i)]$, where F is the cumulative distribution function for the distribution being tested. u_i and l_i denote the upper and lower limits for the i-th class respectively and N is the sample size. This test is sensitive to the choice of bins. There is no optimal choice for bin width, as the optimal bin width depends on distribution, but reasonable choices of bin width should produce similar, not

identical, test results.

The goodness-of-fit chi-squared statistic for testing whether the NPV variable follows the shifted Weibull distribution gives a test statistic of $\chi^2 = 418.3$ with 33 degrees of freedom for ML estimates, where we used integer values between 1 and 35 to produce the bins. The last bin put all NPV ≥ 35 into a single category. As the 0.99-th quantile of the NPV is 35. The p-value is less than 10^{-12} which express that there is a very poor fit on data by the proposed model. This result contradicts the visualizations of the fitted distribution. It seems that basic statistical goodness-of-fit tests do not perform well for our data set. Following this, we calculate the corresponding effect size for the goodness-of-fit test. The results of the Chi-square test is given in Table 3.1.

3.2.5 Effect size for goodness-of-fit

Statistical significance does not necessarily provide information about the importance or magnitude of a measured difference. Instead, many use indicators known as effect sizes (ES), to quantify the importance of such a difference. Cohen (1977) proposed an effect size to measures the magnitude of the discrepancy between a reference distribution and a given distribution in alternative hypothesis based on a table of frequencies. An effect size can be a statistic similar to the chi-square statistic, but it uses the proportions, instead of the frequencies, given as follows:

$$\psi = \sqrt{\sum_{i=1}^{K} \frac{(p_i^o - p_i^e)^2}{p_i^e}}, \tag{3.4}$$

where p_i^o and p_i^e are respectively the observed and expected proportion for the i-th bin. The expected frequency is calculated by $p_i^e = F(u_i) - F(l_i)$, where F is the cumulative distribution function for the distribution being tested. Cohen (1977) also suggests practical rules to interpret ψ, through several hypothetical examples. An ψ around 0.1 is deemed a *small* effect; $\psi \approx 0.3$ is a *medium* effect; and $\psi \approx 0.5$ is a *large* effect.

Table 3.1 summarises the goodness-of-fit test results as well as the corresponding effect size for different estimation methods. Despite the fact that both K-S and Chi-square test for goodness-of-fit significantly reject the hypothesis of having a reasonable fit on data by the Weibull distribution, the effect size shows a very small value for the goodness-of-fit. That is, the fitted Weibull distribution is very close to the empirical distribution of the NPV and can be considered a good fit. This shows that due to a large sample size, non-parametric statistical tests appear very sensitive to any small departure from the assumption on the null hypothesis, even if the departure is not practically significant.

Table 3.1: *Goodness of Fit test results and effect size for goodness-of-fit, for different parameter estimation method*

Estimation Method	(α , λ)	P-value K-S	P-value χ^2	ψ
LS	(0.76, 4.84)	≤ 0.00	≤ 0.00	0.028
ML	(0.87, 5.39)	≤ 0.00	≤ 0.00	0.059
PPCC	(0.72, 4.32)	≤ 0.00	≤ 0.00	0.037

3.3 Analysis of Session Time Duration

Another important attribute of a web session is the length of visit to the website. Logically, we expect a visitor stay longer if she/he finds the website interesting or relevant to their needs. However, spending a fairly large amount of time on the website may not be considered an advantage for an e-commerce website. An inappropriate web design, for example, may cause a customer to spend more time on the website to find the required item, to get the relevant information, fill in the shipment and payment information and generally take a longer time to complete the online shopping process than necessary. As a general rule, it is a good policy to have a target goal which makes sense for the website.

In practice, clickstream data has very little information about visitors as data are limited to just the browsing behaviour on the website. Potential demographic factors that affect website duration, including gender, age, education and occupation, cannot be investigated, except by using the registered users data (Dreze and Hussherr, 2003).

3.3.1 Relationship with Number of pages visited

In addition to the session time duration, an interesting attribute of browsing behaviour is the amount of time people spend on individual pages of a website. The question is whether visitors spend more time on the pages when they visit more pages on the website after finding the required information. In technical terms, we address the relationship between the number of pages visited and session time duration. It is obvious that the more pages a visitor surfs on a website, the more time is spent on that website. However, how these two measures are associated still needs to be determined. A researcher might be interested in investigating how much time a visitor spends on the page, when they have already visited n pages on the website (see box 3.2).

Questions 3.2 How is the duration on a website affected by the number of pages a visitor spends on website? It seems that the time on the pages increases exponentially as the number of pages visited increases.

Hypothesis 3.2 There is a linear relationship between the variables TD and NPV.

Approaches 3.2 Correlation tests; regression models, generalized linear model

In this section we use explanatory data analysis tools to learn more about the type of association between the session time duration attribute and the number of pages visited. We will also examine whether this relationship is affected by other attributes of the session, such as whether a visit is made by domestic visitors, or the presence of functionality features like conversion.

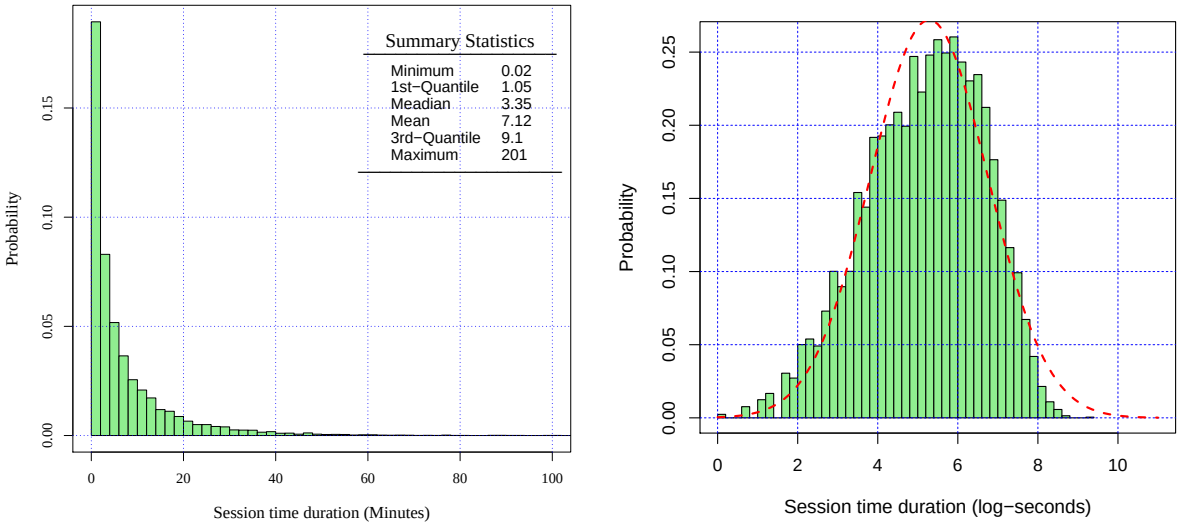
A critical issue about session time spent on a website is that there is no information about the time spent on the last page, and one can only focus on the time duration before the last page viewed. The analyst needs to remove all single-page visits to calculate the measure, as there is no information available on how long they stay on the website.

Graphical representation

First, we have to look at the histogram of the session time duration to learn about the distribution of the variable. Figure 3.7 (left) displays the histogram of the session time duration for sessions during which at least two pages were visited. To increase the granularity, the upper tail has been clipped at 100 minutes. It also includes the summary statistics for the variable TD. The average session time duration is 7.12 Minutes. The session time duration is highly skewed distributed to the right. For this reason, the mean is highly influenced by extreme values and can arguably be used as a measure of central tendency. Therefore, we look at the median as a robust measure of central tendency. The median of the time session duration is about 3.35 minutes, which seems to be a more realistic estimate of the typical session time duration, among those who visited more than one page. It implies that half of the sessions last less than 3 minutes and 20 seconds.

The time duration variables are naturally right skewed. In this situation, it is more common to use the log transformation instead of the original variable to get a distribution closer to the Normal (Mosteller and Tukey, 1977). It is also possible to use the log-Normal distribution if it provides a description of the distribution. For example, Bucklin and

Figure 3.7: *Histogram of the session time duration (left) and its logarithm (right). It should be noted that to avoid negative values of logarithm, time duration has been rescaled to seconds.*



Sismeiro (2003) used a log-Normal model for time spent on web pages. Figure 3.7 (right) shows the histogram for the logarithm of the session time duration. The dashed line shows the fitted Normal distribution based on ML estimate of the parameters. Although, the Normal distribution does not show a perfect fit, it can reasonably approximate the distribution of LogTD.

A scatter plot is a helpful graphical representation for displaying the relationship between two scale variables. Having very skewed variables for both NPV and TD, a scatter plot of the NPV versus TD does not give us a clear picture of the association between two variables, as most of the data points overlap due to lying on a very small range of values. Figure 3.8 (left) shows the relationship between NPV and TD. For better granularity, we concentrate on the sessions which last less than 5 minutes and have less than 15 pages viewed. This graph does not provide a clear association between the two measures. The scatter plot of the log-scaled variables appear in Figure 3.8 (right), revealing the evidence of positive association, between logarithm of the number of pages viewed and logarithm of the session time duration. The correlation coefficient between the two variables is 0.656, indicating a medium strong linear relationship.

Nevertheless the approximate linearity displayed in Figure 3.8 (right) for LogTD against LogNPV shows that a logarithmic transformation is appropriate. For some further insight on the transformation needed see Rao (1973). We perform the regression analysis with a

Figure 3.8: *Scatter plot of the TD versus NPV (left) and logTD versus logNPV including fitted linear regression line (right).*

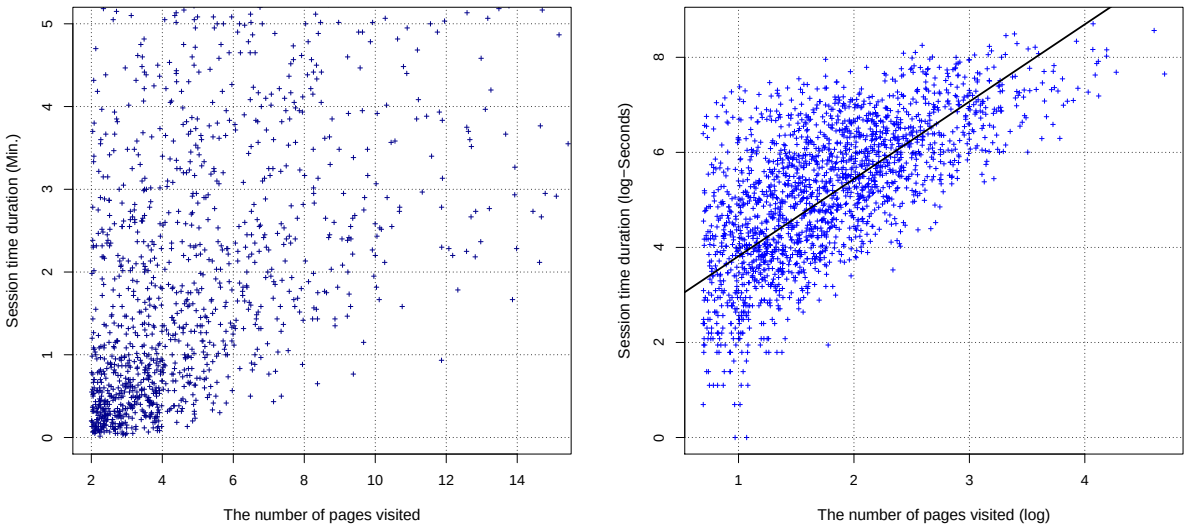
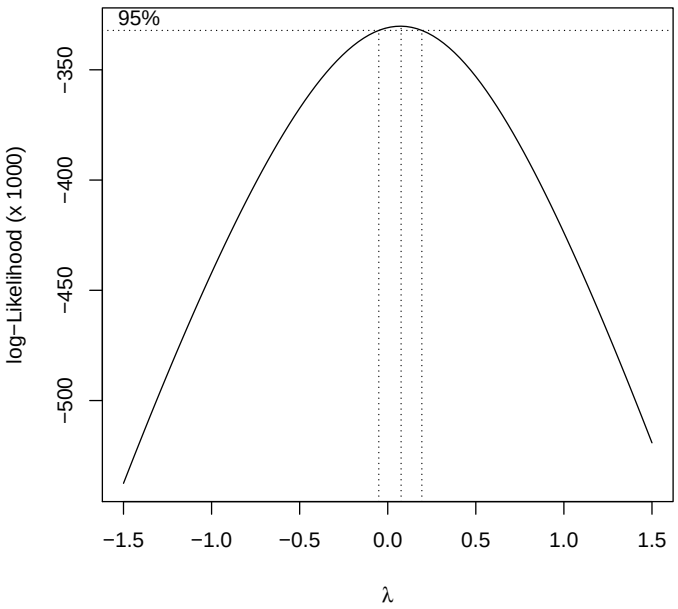


Figure 3.9: *Profile likelihood for a Box-Cox transformation, and 95% confidence band for* λ



general class of transformations of the dependent variable, using the Box-Cox approach (Box and Cox, 1964).

$$y^{(\lambda)} = \begin{cases} (y^\lambda - 1)/\lambda & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{cases} \tag{3.5}$$

As y needs to be positive, the Box-Cox transformation can be applied over $y + \alpha$ for a positive constant α to avoid problems with zero entries. Box and Cox (1964) show that the profile likelihood function for λ is

$$L(\lambda) = c - \frac{n}{2} \log \mathrm{RSS}\big(z^{(\lambda)}\big) \tag{3.6}$$

where c is a constant and $z^{(\lambda)} = y^{(\lambda)}/\dot{y}^{\lambda-1}$, where $\dot{y}$ is the geometric mean of the observations and $\mathrm{RSS}(z^{(\lambda)})$ is the residual sum of squares for the regression of $z^{(\lambda)}$. The Box-Cox method suggests choosing a λ that maximises the profile likelihood function. However, it is desirable to choose easily interpretable transformations such as square-root, log or inverse. We use the function `boxcox` from the package `MASS` to calculate and display the Box-Cox likelihood function, Figure 3.9. The horizontal dashed line provides an approximate 95% likelihood ratio confidence interval for λ. The maximum log-likelihood is obtained at $\lambda = 0.115$. As the confidence band includes the 0, it is more interpretable to use the log transformation. We have also used the `powerTransform` function from package `car` to find the multiple transformation over TD and NPV simultaneously. But, it does not change the model dramatically.

Model estimation and interpretation

This section shows our attempts to use regression models to find the relationship between session time duration and the number of pages visited for the SLC dataset. This analysis may lead to insight about user behaviour in a specific e-commerce website. On the other hand, the model is more descriptive than predictive. The linear regression model assumes that the conditional mean of response variable, Y, given regressors, X, is a linear function of X, whereas the conditional variance of Y given X is a known constant matrix, usually written in the form of

$$\boldsymbol{y} = X\,\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad E(\boldsymbol{\epsilon} \mid X) = \mathbf{0}, \quad V(\boldsymbol{\epsilon} \mid X) = \Omega \tag{3.7}$$

where $\boldsymbol{y}$ is the vector response variable, X is the observed design matrix, $\boldsymbol{\beta}$ is a p-dimentional vector of unknown parameters which needs to be estimated by data, usually

known as *regression coefficients*, and $\boldsymbol{\epsilon}$ is called the error term. The variance-covariance matrix of the vector of error terms is

$$\Omega = \begin{pmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & & \sigma^2 \end{pmatrix}$$

We apply simple linear regression to find the relationship between session logTD and logNPV. The ML estimate of the parameters is given in Table 3.2. So the linear relationship between logTD and logNPV is given by

$$\text{logTD} = 2.73 + 1.37\,\text{logNPV}. \tag{3.8}$$

The statistical test of the null hypothesis $H_0 : \beta_i = 0$ versus the alternative $H_1 : \beta_i \neq 0$ is rejected at a high level of precision for both parameters, p-value$< 10^{-15}$. Exponentiating both sides of the equation 3.8, we can approximate the relationship by the following non-linear form:

$$\text{TD} = 15.3\,\text{NPV}^{1.37}\,. \tag{3.9}$$

The estimated power value 1.37 for NPV, shows the rate by which visiting an extra page increases the amount of time a visitor stays on a website. The coefficient of determination is $R^2 = 0.43$. We also computed the effect size for goodness-of-fit in multiple regression proposed by Cohen (1977), usually known as Cohen's F effect size defined by:

$$\text{Cohen's F} = \frac{R^2}{1 - R^2}\,. \tag{3.10}$$

Cohen's F value in Table 3.2 shows a medium effect size for the association between logNPV and logTD.

Model diagnostics

In this section we use model diagnostic procedures to investigate whether the assumptions of linear regression are satisfied. These procedures include graphical tools as well as statistical tests to check for violation of assumptions such as: error term not having the Normal distribution; variance not being constant across the explanatory variables; fitted relationships being non-linear. For detailed discussion on interpretation of regression diagnostics based on residuals see Draper and Smith (1998)

Table 3.2: *Table of coefficients of the linear regression model for the logTD based on the logNPV and corresponding standard error and p-values.*

Coefficient	Estimate	Std	t-value	Pr($>\|t\|$)
Intercept	2.730	0.0321	73.70	$< 10^{-15}$
Log(NPV)	1.373	0.0166	88.38	$< 10^{-15}$
R-squared=	0.436			
Cohen's F =	0.773			

In the linear regression model it is assumed that the error term follows the Normal distribution. Figure 3.10 shows the histogram and the normal probability plot of the residuals to inspect whether the residuals deviate from the normality assumption. It can be seen that the points in the probability plot form a fairly straight line, indicating that residuals are approximately normally distributed.

Plotting residuals on the y-axis against fitted values on the x-axis is a useful diagnostic tool to check the assumption of constant variance of error term, usually referred to as homoscedasticity. If the model is appropriate, data points appear evenly scattered around the horizontal line at zero. Figure 3.11 (left) suggests that the variance of the residuals decreases as the response value increases, so the model suffers greatly from the heteroscedasticity problem. Furthermore, a statistical test for heteroscedasticity has been developed, as an early effort see Breusch and Pagan (1979). A widely-used test proposed by White (1980), in some literature is the LM test. The LM test statistic is a function of the coefficient of determination, R_e^2, when the squared residuals are considered as a response variable to the original regressors of the model. In practice we fit the squared residuals to the full second-order model implied by the original regressors. For example, for regressors X_1, X_2 we examine the model

$$\epsilon^2 = a + b\,X_1 + c\,X_2 + dX_1X_2 + e\,X_1^2 + f\,X_2^2\,. \tag{3.11}$$

The LM test statistic is given as:

$$LM = n \times R_e^2, \tag{3.12}$$

where n is the number of observation. The LM statistic approximately follows a chi-square distribution with degrees of freedom equal to the number of estimated parameters minus one, under the null hypothesis of homoscedasticity (White, 1980). The LM test statistic and its corresponding p-value are computed for equation 3.8, where the p-value of

Figure 3.10: *The histogram of the residuals of the fitted model (left) and the normal probability plot of the residuals (right).*

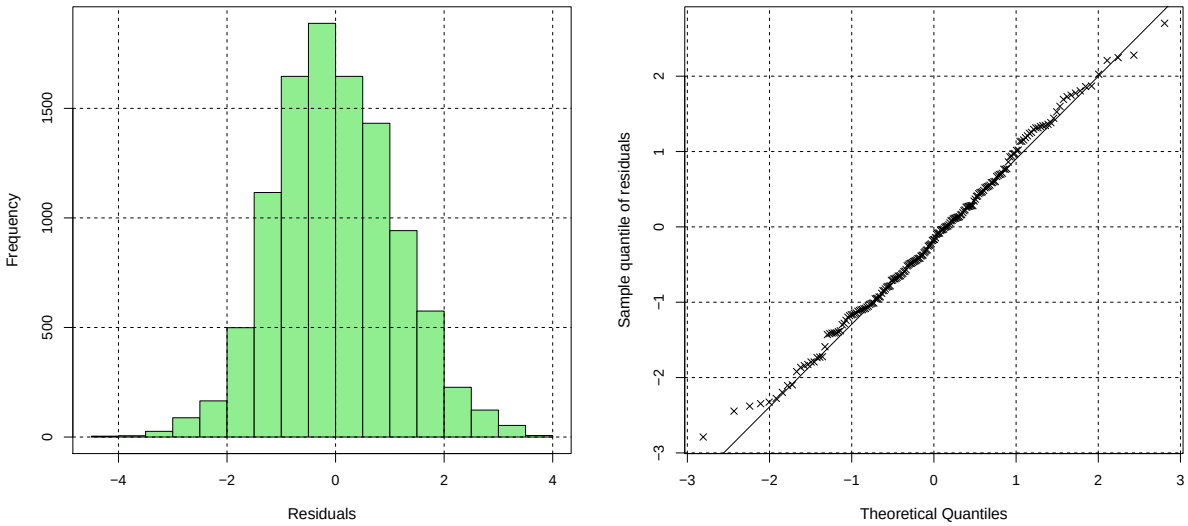


Figure 3.11: *The plot of the residuals versus fitted values (left) and plot of residuals versus order of observations (right).*

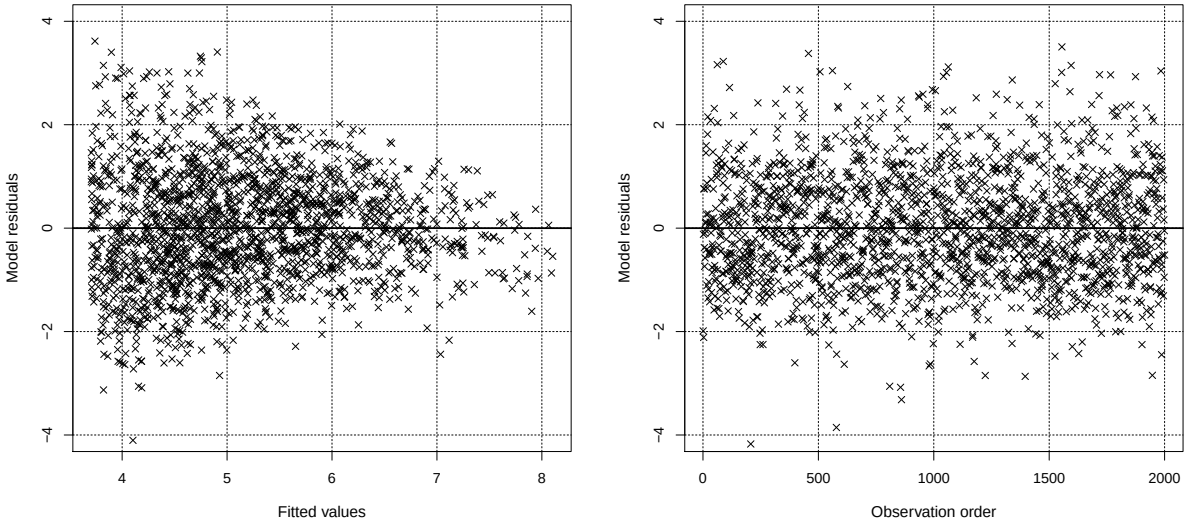


Table 3.3: *Table of coefficients of the linear regression model error and p-values using FGLS estimation, Coefficient of determination, and Cohen's F*

Coefficient	Estimate	Std	t-value	Pr($>\|t\|$)
Intercept	3.363	0.0293	114.09	$< 10^{-15}$
Log(NPV)	1.308	0.0130	100.20	$< 10^{-15}$
R-squared=	0.973			
Cohen's F =	36.032			

$< 10^{-15}$ indicating a critical heteroscedasticity problem of the regression model. It is also necessary to check if observations are independent. This can be graphically observed by plotting the residuals on x-axis versus the order of the observations. Figure 3.11 (right) shows this plot for the fitted model and it does not show any discernible pattern. However, as there exists no natural ordering in our data there is no need to check this assumption.

Feasible Generalized Least Square (FGLS)

In the case of heteroscedasticity, the ordinary least squares can be statistically inefficient, or even give misleading inferences. One solution to unequal variances of the observations is to estimate the unknown parameters in a linear regression model using generalized least squares (GLS). The GLS model assumes less (or no) restriction on variance-covariance matrix of error terms. As a special case, it can be assumed that variance varies for different observations and there is no covariance between errors, so that all the off-diagonal entries of Ω are 0. This special case is also referred to as weighted least square (WLS). That is, the Ω in equation (3.7) is of the following form:

$$\Omega = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_N^2 \end{pmatrix}$$

The GLS estimates of parameters are computed by minimizing the following quadratic from:

$$\widehat{\beta} = \arg\min_{\boldsymbol{\beta}} (\boldsymbol{y} - X\boldsymbol{\beta})' \, \Omega^{-1} (\boldsymbol{y} - X\boldsymbol{\beta}), \tag{3.13}$$

As the true variance-covariance matrix is not known, the GLS estimates of the parameters cannot be estimated directly. One approach is to estimate the variance-covariance matrix and then use it to minimize (3.13). This provides an estimate of parameters which is usually known as feasible generalized least squares (FGLS). Several algorithms have been proposed to find the FGLS. Wooldridge (2010) gives an algorithm which computes the FGLS parameter of coefficients and its variance-covariance matrix in two iterative steps. We use a modified version of this algorithm in the following steps so that the iteration continues until it converges, a small change in the estimates of coefficients:

1. Find the ordinary least square (OLS) estimate of the parameters, and obtain the corresponding residuals,
$$\widehat{\boldsymbol{\beta}}^{\text{ols}} = (X'X)^{-1}X'\boldsymbol{y}. \tag{3.14}$$

2. Find the initial estimate of Ω using the residuals of the OLS estimates. That is, a squared matrix with squared residuals for main diagonal entries and zero for the rest:
$$\widehat{\Omega}^{(1)} = \begin{pmatrix} e_1^2 & 0 & \cdots & 0 \\ 0 & e_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_N^2 \end{pmatrix},$$
where e_i is the observed residual for i-th observation at step 1.

3. Re-estimate the parameters of the model using the initial estimate of the variance-covariance matrix, $\widehat{\Omega}^{(1)}$, and corresponding residuals. The parameters can be computed by
$$\widehat{\boldsymbol{\beta}}^{(t)} = (X'\widehat{\Omega}^{(t)}X)^{-1}X'\widehat{\Omega}^{(t)}\boldsymbol{y}. \tag{3.15}$$
where $t \geq 1$

4. Re-estimate the Ω using the squares of the residuals computed model fitted in step 3. Denoting the residuals by u_i, the variance-covariance matrix is estimated by
$$\widehat{\Omega}^{(t)} = \begin{pmatrix} u_1^2 & 0 & \cdots & 0 \\ 0 & u_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_N^2 \end{pmatrix},$$
where $t \geq 2$. This is called the FGLS estimate of the variance-covariance matrix.

5. Iterate steps 3 and 4 until the estimates of coefficients converge, for example using the Euclidean distance, $\| \widehat{\boldsymbol{\beta}}^{(t+1)} - \widehat{\boldsymbol{\beta}}^{(t)} \| < \epsilon$.

Figure 3.12: *The histogram of the residuals of the fitted model (left) and the normal probability plot of the residuals (right).*

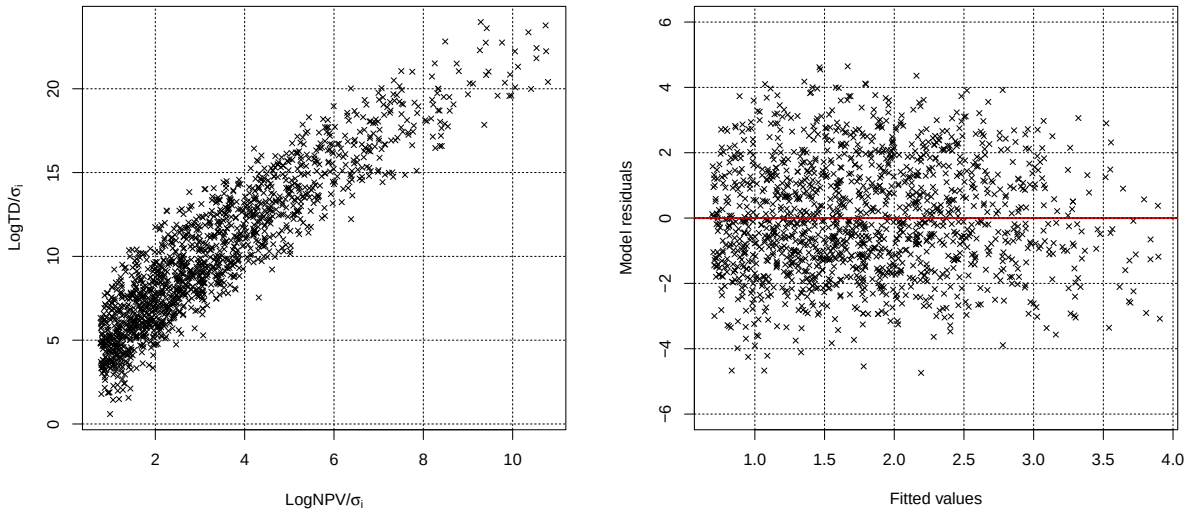
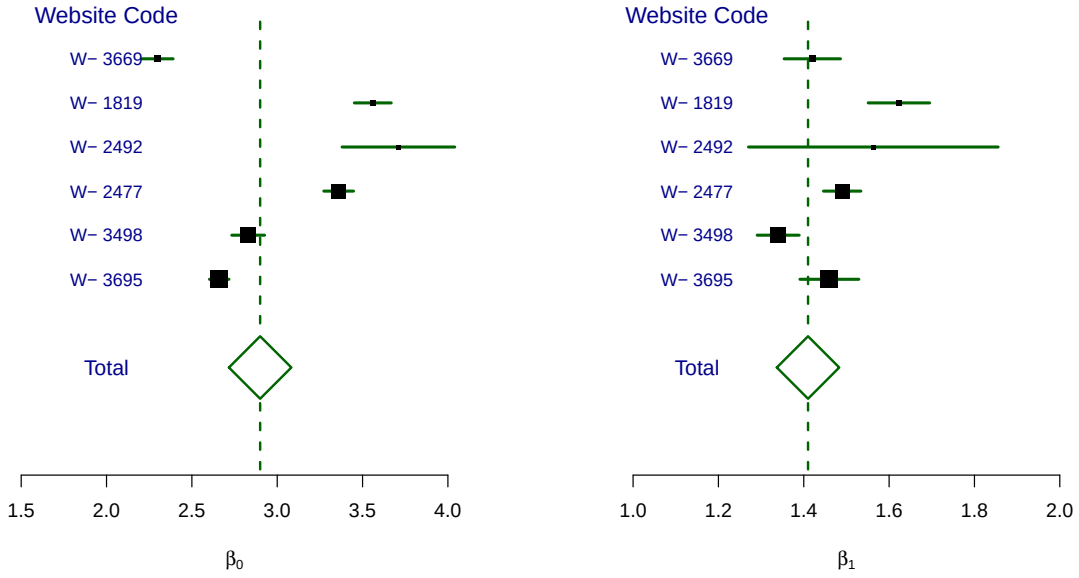


Table 3.3 shows the FGLS estimate of parameters. The estimates do not show a dramatic change compared to the ordinary least squares estimates. However, the estimate of the standard deviation of the coefficient shows smaller values. It should be noted that FGLS can be obtained by fitting the ordinary least square estimate of the linear model through origin, after transforming the response variable and regressors so that it changes into a homoscedastic model. That is,

$$y_i^* = \frac{y_i}{\widehat{\sigma}_i}, \qquad x_{ki}^* = \frac{x_{ki}}{\widehat{\sigma}_i} \tag{3.16}$$

where $\widehat{\sigma}_i$ is the square root of the estimate of variance for the i-th observation, the i-th diagonal element of variance-covariance matrix. The intercept term also transforms into a regressor of $1/\widehat{\sigma}_i$. Figure 3.12 (left) plots the LogTD$/\widehat{\sigma}_i$ versus LogNPV$/\widehat{\sigma}_i$. This plot shows the linear relationship between the dependent and regressor clearly. It can also be seen that the variation is about the same throughout the range of values. The R-squared for no intercept model and corresponding effect size is also represented in Table 3.3 and shows that the FGLS estimate has improved the model - for more information about computing the coefficient of determination for the regression-through-the-origin model see Casella (1983). Figure 3.12 (right) plots the residuals against the fitted values. This plot shows that residuals are about evenly scattered around the horizontal line at zero across the entire range of fitted values.

Figure 3.13: *Forest plot for intercept and slope parameters of the linear regression between LogNPV vs LogTD for several websites*



Meta-analysis of the association

We aim to investigate whether the relationship between the number of pages visited and session time duration given by 3.8 is the same for several websites at the same period of time as SCL data set. We would, of course, have fitted an overarching regression model, and tested for equality of coefficients for different regressors. However, our aim here is simply to assess whether the kinds of result we see seems to replicate for different websites. We compute the FGLS estimate of the parameters of the regression model. The results are summarized using a forest plot to provide a graphical representation of estimates of slope and intercept parameters, as well as their 95% confidence intervals. Forest plots are commonly used in medical research to represent a meta-analysis of the results of randomized controlled trials (Hodges and Olkin, 1985). Figure 3.13 shows the estimated parameters of the model for several website. The estimate of intercept parameter (left), and slope parameters (right) is represented by a square, incorporating 95% confidence intervals represented by horizontal lines. Assuming the normal distribution for the FGLS estimated of parameters, confidence intervals are symmetrical about the estimates for each study. The area of each square is proportional to the number of visit for each website, in our study. The overall measure based on all data from different websites on the plot

is shown by a vertical line and a diamond. If the confidence intervals for individual studies overlap with a vertical line at zero, it demonstrates that at the given level of confidence the hypothesis of no effect is not rejected. The same interpretation is applied for the overall estimate. The graph shows that websites varies significantly in terms of the intercept parameters of the model, range between 2 to 4. The slope parameters are closer in compare to the intercepts around 1.4, showing that the models are in fairly close agreement.

3.3.2 Session time duration for sub-population

In many cases, a researcher is interested in gathering information about two populations in order to compare them with respect to a metric. This is usually known as a two-sample comparison problem. We wish to determine whether two-samples are from the same or different populations. The most familiar statistical test for this situation is the two-sample t-test, used for cases in which two populations may be assumed to be Normal, but with possibly different locations, and with extensions to handle possibly different spreads. In this section we compare the session time duration as a measure of depth of visit for some sub-populations of visitors.

Figure 3.14 (left) shows the back-to-back histogram of the session time duration based on conversion segmentation. The length of a visit is visibly longer for the conversion sessions. The plot, in fact, represents the conditional distribution of the session time duration given the session with and without an online purchase. The graph also shows that in 50% of the non-conversion sessions, the visitor leaves the website within 30 seconds of visiting it. Hence, increase of the session time duration is associated with an increase in the probability of making an online purchase. Figure 3.14 (right) displays the back-to-back histogram for domestic visitors versus non-domestic visitors. For both groups, the distribution of the session time duration seems to be similar, although the summary statistics show that UK visitors spend slightly more time on the site.

The session time duration does not follow the Normal distribution. Hence, the assumption of normality for the t-test does not hold. For this reason, we also compare the sub-population with non-parametric Mann-Whitney Wilcoxon (MWW) test, also known as the Mann-Whitney U test. The MWW test is a non-parametric statistical hypothesis test for assessing whether one distribution is stochastically greater than the other, under the assumption of continuous responses. It is the non-parametric analogue of the two-sample t-test. Table 3.4 shows the test statistics and corresponding p-values of both tests, showing nearly the same result for the significance of the effect on session time duration.

Figure 3.14: *Back-to-back histogram of session time duration given the response variable conversion and non-conversion visits (left); given UK visitors and non-UK visitors (right).*

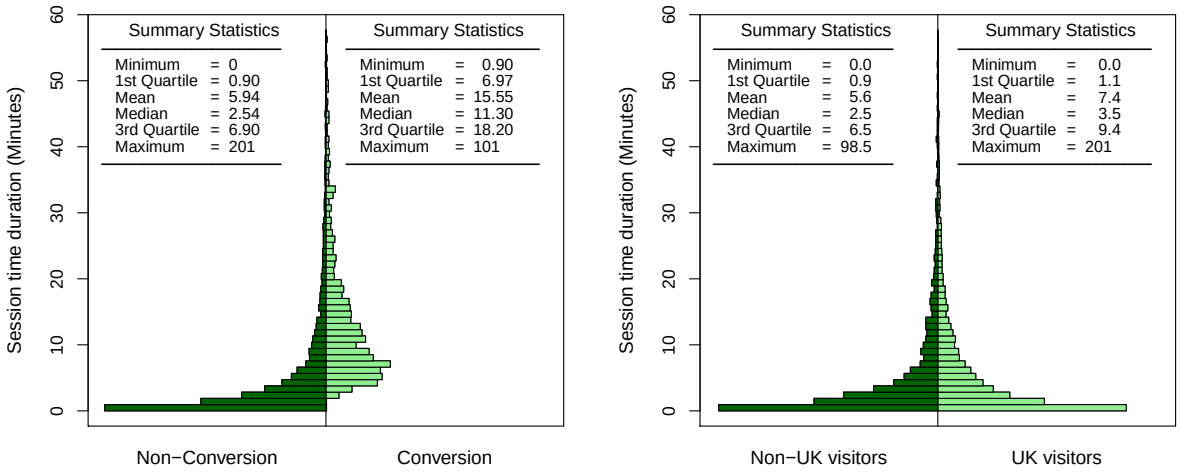


Table 3.4 reports two effect sizes for two-sample comparison studies: Cohen's d and the non-parametric Cliff's δ effect size. Cohen's d (Cohen, 1977) is a widely-used effect size, defined as the difference between the means of two populations divided by their common standard deviation. Cliff (1993) introduced a simple non-parametric index, δ, which is computed by counting the number of occurrences of an observation from one group having a higher response value than an observation from the second group, and vice-versa. The effect size measures for two-sample comparison will be extensively discussed in the next chapter. We bring two effect sizes in Table 3.4 to emphasise again that the result of statistical significant tests do not necessarily show the importance of effect.

The large p-value for the factor *Linked by Google*, 0.49, in Table 3.4 shows that the median of the session time duration for visitors arriving at the website from Google is not different compared to those who directly typed the address. This result is also supported by the small value of the Cliff's δ effect size. This is not the case when we investigate the repeat visit factor for which the p-value corresponding to t-test is significant at any level and the p-value for MWW test is reported 0.011, whilst both effect size measures show small values. So the suggestion is that the conversion and non-conversion group differ in STD, with small p-value and large effect size. However, any other differences are not practically significant in relation to background variation, even though they may be statistically significant.

Table 3.4: *Non-parametric Kruskal-Wallis test to investigate the factors which affect the session time duration. Note that, for factors of two level, the test is equivalent to the Wilcoxon Test*

	MWW Test		Two-sample t-test		Effect size	
Factor	Statistics	p-value	Statistics	p-value	Cohen's d	Cliff's δ
Linked by Google	8690.0	0.496	0.42	0.673	0.01	0.01
Conversion visit	214.2	0.000	26.20	0.000	0.98	0.65
Visit from UK	613.3	0.000	7.45	0.000	0.18	0.11
Repeat Visit	2143.5	0.011	4.47	0.000	0.10	0.03
Visit on Weekend	1036.4	0.061	0.73	0.144	0.03	0.02

Figure 3.15: *Scatter plot logTD versus logNPV marked for two group of conversion and non-conversion visits (left); marked for UK and non-UK visitors (right). It also displays the fitted linear regression line for separate groups.*

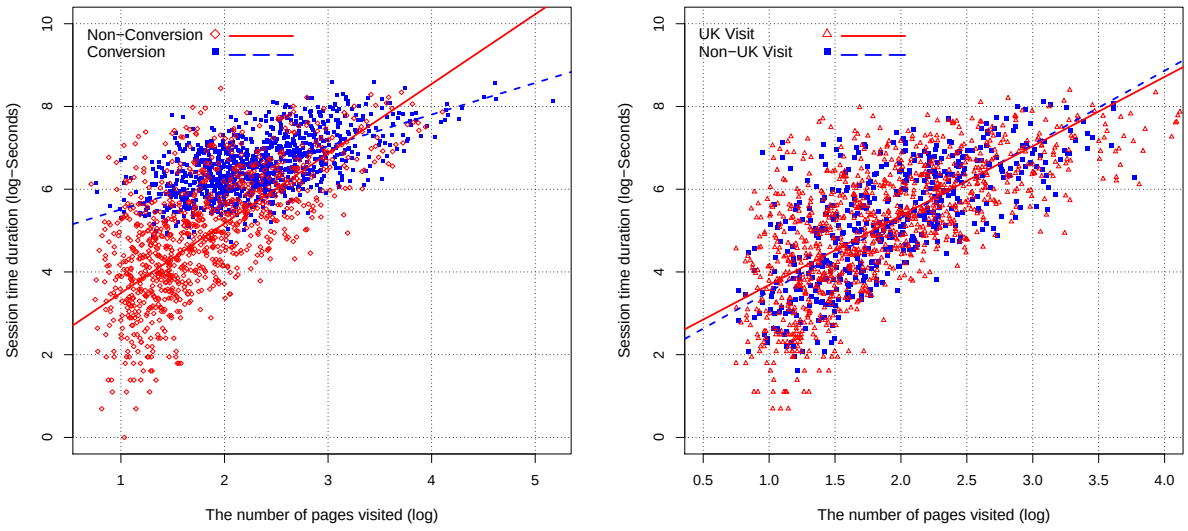
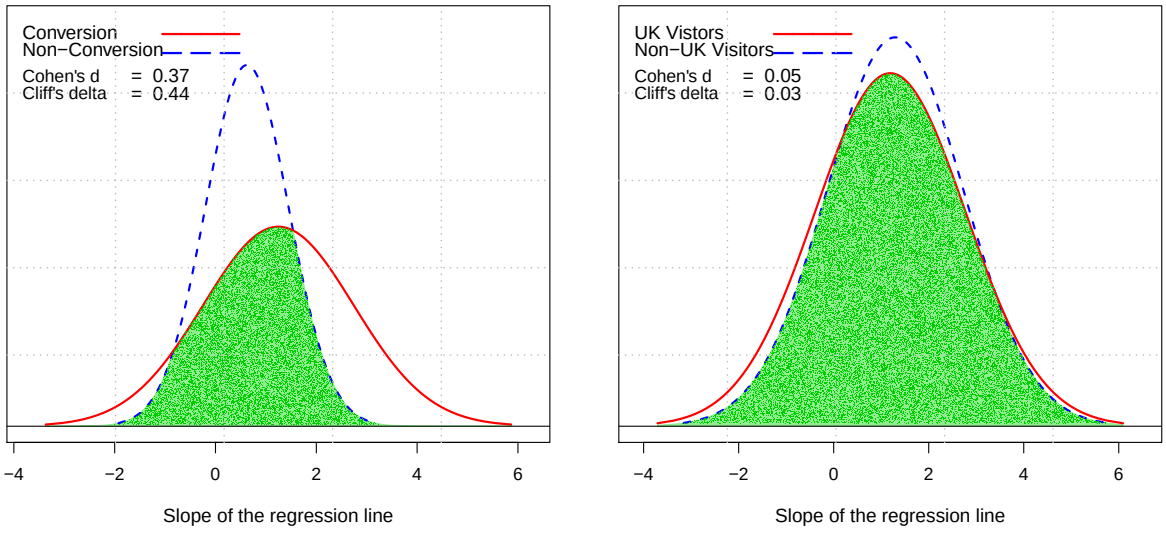


Figure 3.16: *The effect size for difference of the slopes in linear regression line in two groups of conversion sessions and non-conversion sessions (left); and the UK visit group versus non-UK visits (right).*



We also aim to explore whether there is an association between the session time duration and the number of pages visited for different subgroups. As a special case Figure 3.15 (left) shows the overlaid scatter-plot of logTD versus logNPV for both the conversion and non-conversion groups. It also displays the fitted linear regression line separately for subgroups. Similarly, Figure 3.15 (left) shows the scatter plot for the subgroups of UK visits and non-UK visits. It seems that the fitted line between logTD and logNPV has different slope and intercept for the group depending on conversion visits and non-conversion. But, there is no such pattern for UK and non-UK visitors. The bigger slope of the fitted line for the non-conversion than for the conversion group visits shows that session time duration increases more rapidly, as the number of pages visited increases.

It is also desirable to calculate the magnitude of difference between the slope coefficients of fitted linear regression for sub-groups. using the difference between their probability distributions, usually known as a divergence measure. We illustrate the well-known Kullback-Leibler divergence measure in the next chapter, as it can serve as an effect size (Kullback, 1968).

Figure 3.16 (left) provides a graphical representation of the conditional distribution of the estimates of slope parameter given the conversion or Non-conversion visits. It should be noted that we assume the ML estimate of coefficients follow the Normal distribution.

The dashed line represents the distribution of the estimate for the subgroups of visitors without conversion. Equivalently we plot the distribution of the estimate of slope for the subgroup of conversion visits by a solid line. It can be seen that the distribution of the coefficient estimator for the conversion group has a mean (and median) larger than the non-conversion group. Furthermore, the variation of the distribution for conversion group is larger than the non-conversions. The graph shows that distribution of the estimate of slope is very significantly different, whilst the effect size measures of Cohen's d and Cliff's δ show medium effect size. We will discuss the effect size measures in more detail throughout the next chapter. The overlapped area by two distribution is highlighted to represent the similarity (or dissimilarity) of distributions graphically. Figure 3.16 (right) shows the distribution of the estimate of the slope for both subgroups of UK visitors and Non-UK visitors. It can be seen that these distributions are very similar. This is also expressed by the corresponding small effect size measure.

3.4 Discussion

In this chapter we used the explanatory analysis for the attribute of depth of visit for web session, using two measures of the number of pages visited and the session time duration. We showed that the number of pages visited can reasonably be approximated by the Weibull distribution. We illustrated three common ways of estimating the parameters of the Weibull distribution. The goodness-of-fit test strongly rejects the hypothesis that the number of pages visited follows the Weibull distribution. However, the goodness-of-fit effect size and graphical tools reveal a small value for the difference between the fitted and the theoretical distribution. It should be noted that statistical significance does not necessarily provide information about the importance or magnitude of a phenomenon. Instead, one needs to use indicators known as effect sizes (ES) to quantify the importance of such a difference.

Statistical significance is not a direct measure of ES, but there exists a functional relationship between the sample size, the ES and the p-value. For this reason, if the sample size is sufficiently large, even a weak ES may appear as statistically significant. Therefore, a standard statistical test may fail to provide the necessary information in the case of clickstream data analysis when a huge amount of data is available. In the next chapter we introduce two robust effect sizes base on quantile function, which can be used for non-normal circumstances such as depth of visit measures.

Chapter 4

Robust and Scale-free Effect Sizes: Clickstream Analysis

Statistical significance does not necessarily provide information about the importance or magnitude of a measured difference. Instead, many use indicators known as effect sizes (ES), to quantify the importance of such a difference. Statistical significance is not a direct measure of ES, but there exists a functional relationship between the sample size, the ES and the p-value. For this reason, if the sample size is sufficiently large, even a weak ES may appear as statistically significant. The ES has been mainly introduced and investigated for changes in location under an assumption of Normality for the underlying population. However, there are many circumstances where populations are non-Normal, or depend on scale and shape and not just a location parameter. In this chapter, we critically review the common ES measures. We then introduce two novel alternative ES for two-sample comparisons, one scale-free and one on the original scale measurement, and analyse some of their theoretical properties. We examine these ES for two-sample comparison studies under an assumption of Normality and investigate what happens when both location and scale parameters differ. We explore ES for phenomena for non-Normal situations, using the Weibull family for illustration. Finally, for an application, we assess differences in customer behaviour when browsing E-commerce websites.

4.1 Introduction

Classical hypothesis-testing is the standard way of using experimental data to test whether a phenomenon exists (Gigerenzer, 1993; Ledesma et al., 2009). However, statistical signif-

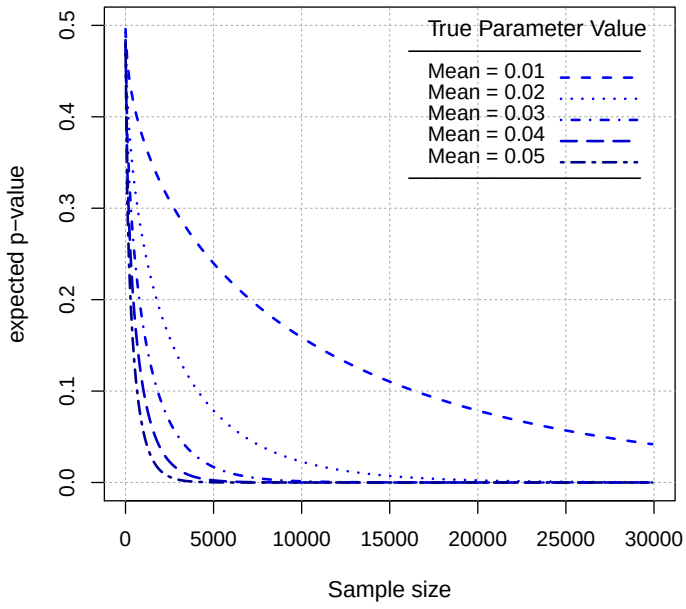
icance does not necessarily provide information about the importance or magnitude of the phenomenon, whereas this is often the focus of researchers in science and social science (Krueger, 2001). For this reason, providing such measures – known as effect sizes (ES) – is considered a necessary complement to hypothesis-testing (Thompson, 1998). This typically concerns four quantities: the p-value, the sample size, a measure of ES, and the power of the test. Each of these quantities is typically a function of the other three (Cohen, 1992; Descôteaux, 2007) and so in quantifying and interpreting experimental research we must consider them together. The p-value is not a direct measure of ES. Statistically significant findings are not always practically significant findings, partly because taking a large enough sample size will result in a small p-value. This is illustrated in Figure 4.1, which plots the p-value from the standard Normal-based test, versus the sample size for the hypothesis test of $H_0 : \mu = 0$ versus $H_1 : \mu > 0$. The expected p-value can be computed by

$$\text{p-value} = 1 - \Phi(\frac{\mu - \mu_0}{\frac{\sigma}{\sqrt{n}}}) \,, \tag{4.1}$$

where n is the sample size, μ is the true mean of the population, μ_0 is the mean of the population based on the null hypothesis, and σ is the standard deviation of the population. In this example we have $\mu_0 = 0$ and $\sigma = 1$. Different lines represent different true values of μ in the population studied, where the values $\mu = 0.01, \ldots, \mu = 0.05$ are so close to zero as to be practically insignificant in most contexts. In each case, we can choose a sample size large enough to generate statistical significance, however tiny the practical significance. Conversely, a statistical test with weak power, perhaps because of small sample size, may not appear as statistically significant, but the measured effect relative to background variation may be deemed to be of practical significance (Wilson Van Voorhis and Morgan, 2007). For this reason, appropriate specification of both statistical significance *and* the magnitude of effect, is required to provide inference of practical utility (Thompson, 1998).

Measures of ES have been available for decades, but mainly limited to meta-analysis for combining estimates from different studies (Keselman et al., 1998). Except for simple situations, they are not found in many statistics computer packages. Hence, even for researchers who are interested in using measures of ES, it is difficult to compute such measures. One reason why ES might not be routinely employed is that they have been largely developed for simplistic situations, such as changes in location under assumptions of Normality. However, there are many circumstances where populations are non-Normal, or depend on scale and shape and not just a location parameter. There do exist some non-parametric measures of ES which don't need such assumptions, but these mostly serve as *dominance* measures. It is important to distinguish between the concepts of ES and dominance. We regard one distribution F as *dominating* a second distribution G when

Figure 4.1: *The p-value of the test versus the sample size for* $H_0 : \mu = 0$ *versus* $H_1 : \mu > 0$, *given the true value of* μ *for some* $N(\mu, 1)$ *examples.*



each quantile of F is larger than the corresponding quantile of G. However, there are many settings in which two distributions do not necessarily dominate each other, but exhibit contrasting differences for different parts of the range. For example, when comparing two lifetime distributions for a survival analysis, the survival function for one group may be initially larger than that for the other group, but then cross over and become smaller. For such cases, classical ES do not truly describe these differences.

In this chapter, we briefly review existing ES measures for two-sample comparison studies. We then introduce two robust ES measures based on quantile differences and use these to examine changes in location under a Normality assumption, and thence to changes in location and scale. We show how the proposed ES behaves for general comparisons of two distributions. For a practical application, we compute ES in the context of identifying patterns in web browsing behaviour.

4.2 Effect sizes for the two-sample comparison

4.2.1 Cohen's d and d_r

The most familiar ES is Cohen's d (Cohen, 1977), defined as the difference between the means of two populations divided by their common standard deviation. In practice, d is estimated by $d = (\bar{x}_1 - \bar{x}_2)/s$, where $\bar{x}_1$, $\bar{x}_2$ are the sample means, and s is an estimate of the common standard deviation. When the population standard deviations cannot be assumed to be the same, Hedges (1981) proposed using the usual pooled standard deviation s_p estimate to replace s, where $s_p^2 = [(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2]/(n_1 + n_2 - 2)$. Cohen's d is one of the most widely-used ES measures for comparing the means of two independent samples and expresses the intuitively appealing concept that the magnitude of effect is the difference between the centres of two populations relative to a measure of individual variation. There is often implicitly the assumption that the underlying populations are Normal.

Cohen (1977) gave practical rules to interpret ES. An ES of 0.2 to 0.3 standard deviations is deemed a *small* effect; ES $\approx$ 0.5 is a *medium* effect; and ES $\approx$ 0.8 is a *large* effect. He warned that such criteria are relative and interpretations must take into account the content, purpose, and method of research, except perhaps in the context of research with entirely novel variables (Lenth, 2001). The value of this rule to applied research has been questioned, since the practical importance of ES also depends on other quantities, such as the effectiveness of alternative treatments and cost-benefit analysis of the treatment

Hodges and Olkin (1985).

Cohen (1977) also proposed a correlation form, d_r, of his d index, where d_r is the correlation between numerical variable Y and binary variable X. Cohen's d_r is

$$-1 < d_r = \frac{d}{\sqrt{d^2 + (1/pq)}} < 1, \tag{4.2}$$

where p and q are the proportions of subjects belonging to groups A and B (usually the experimental and control groups) respectively, as indicated by X. Cohen's d_r is large when d is large and also when p and q are similar. d_r is a bounded index which facilitates interpretation of the ES and this might be considered an advantage of using d_r instead of d.

4.2.2 Robustness

There have been many attempts to extend Cohen's d to more complicated situations, for example weakening the assumption of common variance. Glass et al. (1981) initially recommended using $(\bar{x}_1 - \bar{x}_2)$ scaled by the estimated standard deviation of the control group instead of s_p. However, this simply ignores that we might prefer an ES which explicitly incorporates heteroscedasticity. Typically, Cohen's d and other commonly used ES are not robust: small changes in the tails can substantially inflate variance estimates and thus decrease Cohen's d. A well-known example is based on the contaminated Normal distribution (Tukey, 1960; Wilcox, 2005; Wilcox and Tian, 2011). Suppose that $H(x) = 0.9\Phi(x) + 0.1\Phi(x/10)$, where Φ is the standard Normal cdf. Now consider two groups, both Normally distributed and with common variance $\sigma^2 = 1$, and suppose their means are $\mu_1 = 0$, $\mu_2 = 0.8$, so that $d = 0.8$, a *large* ES. For the contaminated Normals with the same means, Cohen's d falls to $d = 0.59$, a *medium* ES.

In general, replacing the mean and variance with some robust estimate can be used to give a robust ES; for example Algina et al. (2005) uses 20% trimmed means and a Winsorized variance. There is a loss of tail information when trimming. Hedges and Friedman (1993) proposed an analogue for Cohen's d when the focus is on comparing the tails or some other feature of a distribution rather than the centre. For example for the upper tail of the distribution, $d_\alpha = (\bar{x}_{1\alpha} - \bar{x}_{2\alpha})/s_\alpha$, where $\bar{x}_{j\alpha}$ for $j = 1, 2$ are the means of observed values higher than the α-quantile and s_α their (assumed) common standard deviation.

4.2.3 Common language effect size

The Common Language Effect Size (CLES) statistic is a probability measure generating an ES when comparing two means of independent samples. It is defined as the probability that a randomly selected individual from one group has a higher score on a variable than a randomly selected individual from another group (McGraw and Wong, 1992). Thus, if X and Y are independent Normal $N(\mu, \sigma^2)$ quantities, the CLES is:

$$l_1 = P(X < Y) = \Phi(\frac{\mu_x - \mu_y}{\sigma\sqrt{2}}). \tag{4.3}$$

A value of $l_1 = 0.5$ implies that two distributions entirely overlap. Values close to zero or one imply a larger ES and less overlap. The CLES can be calculated for different probability distributions and under different assumptions and has the advantage of representing ES using a common probability scale (Ledesma et al., 2009). CLES is often employed as a measure of numerical dominance in location to judge whether one distribution is generally larger or smaller than another. For example, $l_1 = 0.5$ for Normal random variables with the same mean, irrespective of variance.

Anderson and Berry (2009) proposed a modified version of the CLES for the cases for which the population follows a distribution with both location and scale, and for which both parameters may differ. In this case, if the standardized random variables $\frac{X-\mu_x}{\sigma_x}$ and $\frac{Y-\mu_y}{\sigma_y}$ each follows the distribution of some standard random variable Z symmetric around 0, with cumulative distribution function Φ_Z, a dominance measure can be computed as:

$$l_2 = \Phi_Z \left(\frac{\mu_x - \mu_y}{\sigma_x + \sigma_y} \right). \tag{4.4}$$

A corresponding point of overlap is given by

$$\frac{\sigma_y \, \mu_x + \sigma_x \, \mu_y}{\sigma_x + \sigma_y}, \tag{4.5}$$

which is familiar as the posterior mean in various Bayesian statistical settings.

The probability of superiority ES (Grissom and Kim, 2005) is similar to the CLES, but based on sample ranks and given by $PS_U = U/n_1 n_2$, where U is the Mann-Whitney U statistic . This measure assumes similarity of shape in the underlying populations, but is not robust to heteroscedasticity (Mann and Whitney, 1947; Grissom, 1994).

4.2.4 Non-overlap effect size

An ES for the two-sample comparison can be obtained by comparing the percentiles of the populations. Cohen (1977) introduced three such measures under an assumption of

Normality with equal variation. One measure is the probability that random variable X is less than the mean of random variable Y:

$$u_1 = P(X < \mu_y) = \Phi(\frac{\mu_y - \mu_x}{\sigma}). \tag{4.6}$$

Cohen's second measure gives the overlap formed by taking the percentage in one density that exceeds the same percentage in the other density. For Normal distributions with equal variance this is the probability that the distribution of one population lies below the joint average:

$$u_2 = P(X < \frac{\mu_x + \mu_y}{2}) = \Phi(\frac{\mu_y - \mu_x}{2\,\sigma}). \tag{4.7}$$

Cohen's third measure considers the part of the two densities that do not overlap. This can be expressed as $u_3 = 2 - 1/u_2$.

These non-overlap ES are probability measures and so can be applied without reference to context. They can be used to help interpret Cohen's d in circumstances when the variable under study is new and when its behaviour has not been widely studied – for example, in psychological studies when a new questionnaire has been introduced to measure an characteristic. Thus, assuming Normality, when $d = 0.5$, the overlap area is 23% and correspondingly $u_3 = 0.67$ (Ledesma et al., 2009).

4.2.5 Non-parametric effect size

Cliff (1993) introduced a simple non-parametric index, δ, which is computed by counting the number of occurrences of an observation from one group having a higher response value than an observation from the second group, and vice-versa. This statistic estimates the probability that a value selected from one group is greater than a value selected from the other group, minus the reverse probability. Thus,

$$\delta = \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\text{sign}(x_{i1} - x_{j2})}{n_1 \times n_2}, \tag{4.8}$$

where x_{i1} is the i-th observation from group A, and x_{j2} the j-th observation from group B; and n_1 and n_2 are the respective sample sizes. $\delta = \pm 1$ indicates complete separation between the two groups, whereas $\delta = 0$ indicates complete overlap.

When the data do not follow the Normal distribution or where the variable under study corresponds to an ordinal level of measurement, Cliff's δ might be preferred to Cohen's d or d_r (Hess and Kromrey, 2004). Cliff's δ is a non-parametric measure, so its interpretation is unaffected when the assumptions of Normality or homoscedasticity are violated (Coe, 2002). However, δ is more useful as a crude dominance measure.

4.2.6 Explanatory power effect size

Wilcox and Tian (2011) proposed an approach to measuring ES based on special case measures that reflect the proportion of variance in a response variable Y accounted for by an explanatory variable X using regression. Doksum and Samarov (1995) gave such a measure for simple linear regression studies. One advantage of this approach is that it can be generalized to multi-sample problems. There are numerous robust analogues based on robust regression techniques. For two-sample studies, X takes values 0 and 1 corresponding to the two groups. See Kulinskaya and Staudte (2006) for a suggested measure, but note that their measure depends on sample size and so can't be fully recommended.

4.2.7 Graphical representations

Quantile-quantile and percentile-percentile plots have long been exploratory graphical tools for two-sample comparison studies (Wilk and Gnanadesikan, 1968). Doksum and Sievers (1976) and Doksum (1977) suggested using a graphical representation of the shift function for comparing two groups. This involves making comparisons at a number of distribution features, rather than only location or scale parameters. The magnitude of difference between two populations may also be explored using the ordinal dominance curve (Darlington, 1973), where the area under or above the ordinal dominance curve may be used as a measure of dominance. For a better separation of points when observations are close to the $y = x$ line, the Tukey sum-difference graph has been recommended instead of the pp plot and qq plot (Cleveland, 1994). Such plots provide a complete representation of the differences between two populations, but of course don't offer a single summary ES.

4.3 Developing effect sizes for non-Normal data

There are many circumstances where populations are known or suspected to be non-Normal, and perhaps depend on scale and shape as well as location. Basing an ES on medians alone does not help: for example, the medians of two distributions might be equal, but the tails of the two distributions might be very different (Fleming et al., 1980). Indeed, where two populations may be Non-Normal, or heteroscedastic, or nonhomomerous in other ways, traditional measures of ES can be misleading and do not provide sufficient information about the magnitude of the effect (Grissom and Kim, 2005; Wilcox, 2005). However, it remains desirable to have a measure of effect size which is credible for such settings, especially if such effect sizes can be computed on a scale-free basis in order to

facilitate comparisons from several studies, for example as further inputs to meta-analysis.

Our need is to construct an ES over the full distribution. Extending from the notion of the graphical comparison described above, one possibility for the two-sample comparison is to compare the *quantile functions* and *vertical quantile comparison functions* of two distributions F and G. In principle, these should provide general ES which are applicable to both non-Normal and Normal settings.

Definition: Quantile function. Let F be a probability distribution function. The quantile function for F is given by

$$Q(p) = \inf \{x \in R : p \leq F(x)\}, \quad 0 \leq p \leq 1. \tag{4.9}$$

For discrete probability distribution functions, the Quantile function returns the minimum value of x for which the statement holds. For random variables for which the cumulative distribution function (cdf) is continuous and strictly monotonic, $F \colon R \to (0, 1)$, $Q(p)$ is the inverse of the cdf, and it is common to use F^{-1} as notation.

Definition: Vertical quantile comparison function. Suppose that F and G are continuous probability distribution functions. The vertical quantile comparison function for G with baseline F is:

$$V_F^G(p) = G(F^{-1}(p)), \quad 0 \leq p \leq 1. \tag{4.10}$$

This may be used to represent the distance between two probability distributions (Li et al., 1996). If $F = G$ then $V_F^F(p)$ is the probability distribution function of the Uniform distribution. Thus, differences between $V_F^G(p)$ and the Uniform cdf equate to differences between F and G. One may plot the vertical quantile comparison function versus the probability to display the difference between two distributions (Holmgren, 1995).

Definition: Vertical shifted function. The vertical shifted function for G with baseline F is defined to be $V_F^G(p) - p$. This summarizes differences between F and G at each point p because the Uniform cdf is $F(p) = p$.

4.3.1 Quantile absolute deviation

In this section, we extend the idea of comparing two distributions by their quantiles for the entire range of probabilities over $[0, 1]$.

Definition: Quantile Absolute Deviation. For two populations with cumulative

distribution functions F and G, we define the Quantile Absolute Deviation (QAD) as:

$$K(F\ ,\ G) = \int_0^1 \Big|\ F^{-1}(p) - G^{-1}(p)\ \Big|\ dp, \tag{4.11}$$

where F^{-1} and G^{-1} are quantile functions for the two distributions. The QAD is a symmetric positive measure and may be interpreted as the average distance between the quantiles of the distributions. When two distributions are similar, their quantiles must also be similar and $K(F\ ,\ G)$ will be small. The QAD satisfies the three divergence properties of a criterion:

1. Self similarity: $K(F\ ,\ F) = 0$.
2. Self identification: $K(F\ ,\ G) = 0$ if and only if $F = G$.
3. Positivity: $K(F\ ,\ G) \geq 0$ for all F, G.

The space of quantile functions with the above mentioned distance is a metric space because the distance fulfills all the following axioms of a metric. For probability distribution functions F, G, and H:

1. Self identification: $K(F\ ,\ G) = 0$ if and only if $F = G$.
2. Symmetry: $K(F\ ,\ G) = K(G\ ,\ F)$.
3. Triangle inequality: $K(F\ ,\ G) + K(G\ ,\ H) \geq K(F\ ,\ H)$.

The QAD is not a scale-free measure as it has the same unit of measurement as the variable under investigation. Thus, $K(F\ ,\ G) = 1$ implies that on average the quantiles of the variable with distribution F differ by one unit from the quantiles of distribution G. It may or may not be true that one of the distributions dominates the other, but if this is the case, the QAD may be interpreted directionally.

One may exclude the observations which lie in the far tails of the distribution, in order to eliminate the effect of outliers or extreme values on the ES. A $100\alpha\%$ trimmed QAD can be calculated using only observations which are placed between the $(\alpha/2)$-th and $(1 - \alpha/2)$-th percentiles of the data. This measure can be computed as:

$$K_\alpha(F\ ,\ G) = \frac{1}{(1-\alpha)} \int_{\alpha/2}^{1-\alpha/2} \Big|F^{-1}(p) - G^{-1}(p)\ \Big|\ dp. \tag{4.12}$$

For illustration, a $K_{0.1}(F\ ,\ G)$ is computed by discarding the lower and higher 5% tails of each of the two populations and by calculating the rescaled $K(F\ ,\ G)$. $K_\alpha(F\ ,\ G)$ is more

robust to outliers, but does not give a metric measure over the space of quantile functions and loses tail information.

4.3.2 Quantile comparison effect size

In this section, we introduce the notion of using a distance measure – statistical divergence – as an ES. Statistical divergence is a weaker notion than that of distance as it does not need to be symmetric. That is, the divergence from F to G is not necessarily equal to the divergence from G to F. The vertical shifted function $V_F^G(p) - p$ provides a basis for such a divergence. For two cumulative distribution functions F and G, define

$$Div(F \parallel G) = 2 \times \int_0^1 \Big| G(F^{-1}(p)) - G(G^{-1}(p)) \Big| dp = 2 \times \int_0^1 \Big| \ V_F^G(p) - p \ \Big| dp, \quad (4.13)$$

which takes values over the interval $[0, 1]$. Although this measure satisfies divergence properties, it is not a symmetric measure as $Div(F \parallel G) \neq Div(G \parallel F)$ necessarily. Instead we may obtain a symmetric measure by averaging these divergences, and this is what we propose to use as an ES.

Definition: Quantile comparison effect size. We define the Quantile comparison effect size (QCES) as

$$\begin{aligned} D(F\ ,\ G) &= \frac{1}{2} Div(F \parallel G) + \frac{1}{2} Div(G \parallel F) \\ &= \int_0^1 \Big| \ G(F^{-1}(p)) - p \ \Big| + \Big| \ F(G^{-1}(p)) - p \ \Big| \ dp. \end{aligned} \quad (4.14)$$

This is a bounded measure, giving values between 0 and 1, which facilitates its interpretation as an ES, unlike alternative unbounded divergence measures.

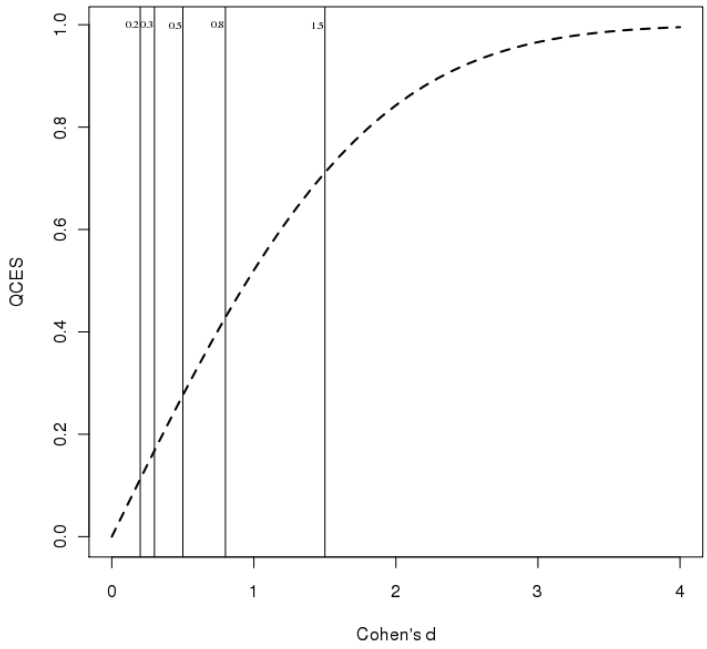
There are, of course, alternative divergence measures such as the Kullback-Leibler (KL) divergence. This is the relative entropy between two continuous probability density functions $f(x)$ and $g(x)$ (Kullback, 1968):

$$D_{KL}(D \parallel G) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} dx. \quad (4.15)$$

This is commonly used as a measure of similarity between two density distributions. In a Bayesian context, this divergence and its variants may be used to measure the difference between prior and posterior distributions, and symmetrized versions may be obtained.

ES based on quantile comparison can be computed directly where F and G are known. Otherwise we may substitute the cdf by the empirical cdf. Thus, suppose we draw samples

Figure 4.2: *Values of the QCES corresponding to standard thresholds for Cohen's d ES.*



from distribution F and distribution G and estimate F by $\widetilde{F}$ and G by $\widetilde{G}$, the corresponding empirical cdfs. From these we may determine the empirical quantile functions $\widetilde{F}^{-1}$ and $\widetilde{G}^{-1}$ as appropriate. See the appendix A for algorithms to compute empirical ES.

4.3.3 Interpreting the QC effect size

Cohen's practical rules to interpret ES can be mapped into the QCES by evaluating the QCES for Normal distribution comparisons where we fix the scale at $\sigma_X = \sigma_Y = 1$ and consider location changes of sizes deemed by Cohen to be *small* ($\mu_X - \mu_Y = 0.2$), *medium* ($\mu_X - \mu_Y = 0.5$), and *large* ($\mu_X - \mu_Y = 0.8$). A graph showing the QCES for general choices for $d = \mu_X - \mu_Y$ in this case is shown in Figure 4.2.

If we adopt similar thresholds, this suggests that QCES values of around 0.1 to 0.2 correspond to small effects; values around 0.2 to 0.4 correspond to *medium* effects; and larger values suggest *large* effects. A QCES of at least 0.7 represents the situation where two Normal distributions with the same scale mostly do not overlap. For more complicated comparisons with non-Normal distributions and possible changes in the values of several parameters, distributions may still contain substantial overlap with respect to quantiles. Having assessed the QCES over several plausible comparisons, we thus suggest the follow-

ing guidelines. A value of QCES in (0.10,0.20) suggests a *small* effect; (0.20,0.30) suggests a *medium* effect; and 0.30 and above correspondes to a *large* effect. As Cohen suggests, such ES need to be interpreted in context; computation of a summary ES cannot replace detailed comparison of the two distributions. For the robustness illustration of §4.2.2 using contaminated Normal distributions, the corresponding values of the QCES are 0.26 and 0.21, indicating *medium* effect sizes on our scale in both cases.

4.3.4 ES computation for some simple examples

The computation of QCES and QAD involves integration over a function of two quantile functions. As such, they rarely provide tractable analytic forms, particularly for distributions for which there is no simple expression for the quantile function; see Gilchrist (2000) for details of quantile functions for a large number of known distributions. However, it is informative to investigate the behaviour of these ES in terms of the parameters of distributions in some simple settings. Thus, here we compute QAD and QCES (1) when comparing Uniform distributions with different location parameters; (2) when comparing Exponential distributions with different rate parameters.

Uniform distribution: Suppose that $X \sim U(0\ ,\ \alpha)$ and $Y \sim U(0\ ,\ \beta)$ where $\beta \geq \alpha$. The quantile function for $X \sim U(a\ ,\ b)$ is $F_X^{-1}(p) = (b-a)\,p + a$. So, using (4.9) the QAD is given by

$$K(X\ ,\ Y) = \frac{\beta - \alpha}{2}, \tag{4.16}$$

In this case, the QAD is simply the difference between the means of the two distributions. The QCES (4.14) is computed using $Div(X \parallel Y)$ and $Div(Y \parallel X)$, remembering the restriction to $\beta \geq \alpha$, giving

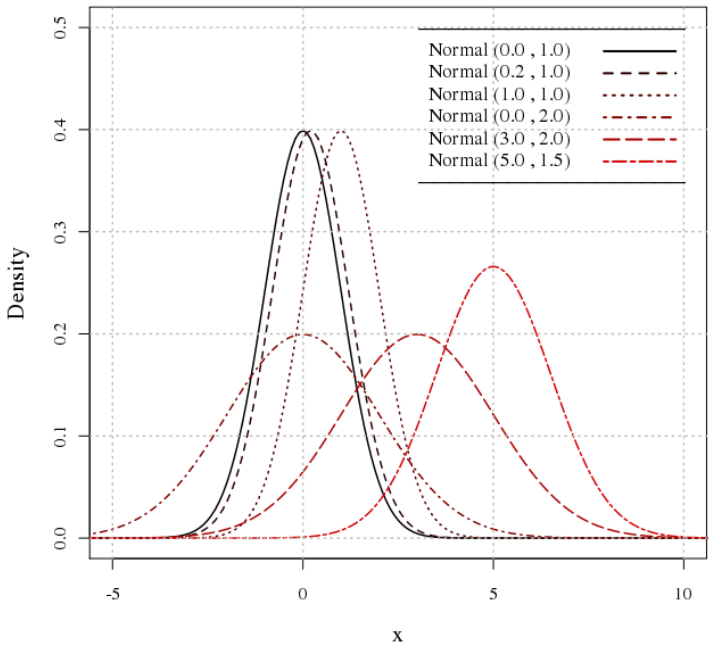
$$D(X\ ,\ Y) = 1 - \frac{\alpha}{\beta}. \tag{4.17}$$

For β close to α the QCES tends to zero, and for $\beta \gg \alpha$ the QCES approaches the upper limit, one.

In practice, we typically won't know the values of parameters and so must estimate them. One way to explore the behaviour of these ES is to treat these parameters as random variables, in which case these ES are random variables. Suppose, for example, that the parameters are independently $\alpha \sim N(\mu_\alpha, \sigma_\alpha^2)$ and $\beta \sim N(\mu_\beta, \sigma_\beta^2)$. In the above Uniform case, the QAD is Normal:

$$K(X\ ,\ Y) \sim N\Big(\frac{\mu_\beta - \mu_\alpha}{2}\ ,\ \frac{\sigma_\beta^2 + \sigma_\alpha^2}{4}\Big) \tag{4.18}$$

Figure 4.3: *The pdfs for the compared Normal distributions.*



suggesting that we would expect the ES to be the underlying difference between the two Uniform means, but with variation stemming from uncertainty about those means. The QCES is a simple linear transformation of a ratio of two independent non-zero mean Normal distributions, as discussed by (Hinkley, 1969). In some cases this ratio distribution may be well approximated by another Normal distribution which can serve as the basis for exploration of behaviour.

Exponential distribution: Suppose that $X \sim Exp(\alpha)$ and $Y \sim Exp(\beta)$ where $\beta \geq \alpha$. The quantile function for $X \sim Exp(\theta)$ is $F_X^{-1}(p) = -\frac{1}{\theta}\ln(1-p)$. Thus, the QAD (4.11) is

$$K(X\ ,\ Y) = \frac{1}{\alpha} - \frac{1}{\beta}, \tag{4.19}$$

being the difference between the means of the two distributions. We find that the QCES (4.14) is

$$D(X\ ,\ Y) = \frac{\beta - \alpha}{\alpha + \beta}, \tag{4.20}$$

so that QAD can be interpreted as the difference in means scaled to (0,1), as is so for the Uniform case. As for the Uniform case, one might be interested in exploring these ES as random variables, assuming Normal distributions for the α and β parameters. However, these distributions are analytically intractable.

4.4 Two-sample Normal distribution comparisons

We now explore the QAD and the QCES for a number of situations. First, we assume that the underlying distributions are known and we examine ES for the Normal distribution comparison and, later, for the Weibull distribution comparison, varying parameter choices in each case. In a further section we examine ES calculations for a practical example for which the underlying distributions are estimated by empirical cdfs.

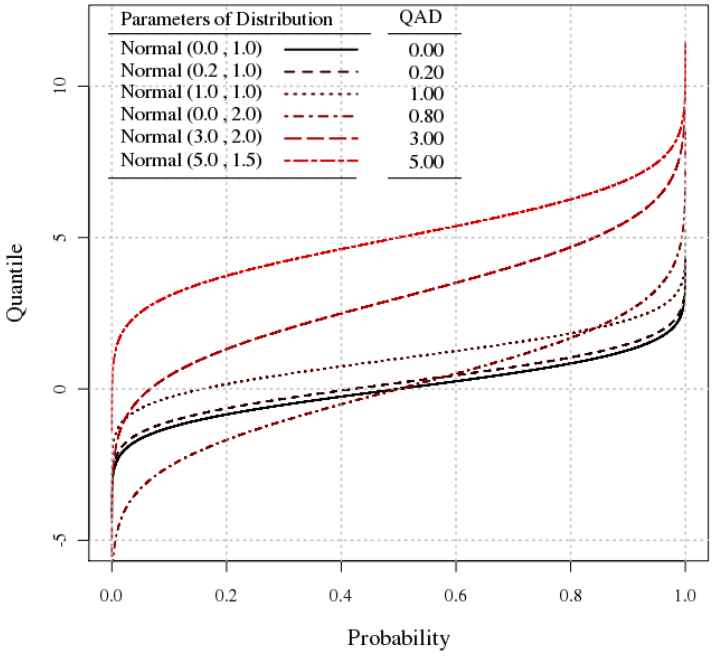
We compare five Normal $N(\mu, \sigma^2)$ distributions to the standard Normal $N(0, 1)$ distribution, as baseline. Figure 4.3 displays them. The parameter choices are the baseline itself, and the $N(0.2, 1)$, $N(1, 1)$, $N(0, 2)$, $N(3, 2)$, and $N(5, 1.5)$ distributions. These give a range of parameter choices for varying mean and standard deviation, and for a range of classical ES; for example: the comparison between $N(0, 1)$ and $N(0.2, 1)$ corresponds to Cohen's $d = 0.2$, a *small* ES. We include a check on the baseline comparison with itself.

Table 4.1: *Mean and standard deviation of Monte Carlo simulations of Cohen's d, Cliff's δ, the QAD and QC Effect sizes, and the KL divergence, for Normal distribution comparisons with $N(0, 1)$ baseline.*

		Effect Size				
		Cohen's d	Cliff's δ	KL	QAD	QCES
$N(0, 1.0)$	mean	-0.002	-0.005	0.133	0.135	0.071
	sd	(0.141)	(0.078)	(0.074)	(0.077)	(0.045)
$N(0.2, 1.0)$	mean	0.200	0.110	0.199	0.220	0.122
	sd	(0.144)	(0.076)	(0.101)	(0.119)	(0.068)
$N(1.0, 1.0)$	mean	1.007	0.521	0.929	0.996	0.521
	sd	(0.149)	(0.061)	(0.198)	(0.144)	(0.066)
$N(0.0, 2.0)$	mean	-0.002	-0.002	1.354	0.810	0.214
	sd	(0.142)	(0.085)	(0.363)	(0.127)	(0.028)
$N(3.0, 2.0)$	mean	1.913	0.819	6.022	2.992	0.821
	sd	(0.186)	(0.041)	(1.180)	(0.222)	(0.041)
$N(5.0, 1.5)$	mean	3.940	0.994	13.256	5.002	0.994
	sd	(0.252)	(0.005)	(2.226)	(0.182)	(0.003)

Table 4.1 shows the mean and standard deviation of Monte Carlo simulations of Cohen's d, Cliff's δ, the QAD and QC Effect sizes, and the KL divergence, for Normal distribution comparisons with a $N(0, 1)$ baseline. For each paired comparison we obtain a random sample of 100 observations from each distribution and then compute ES for the compari-

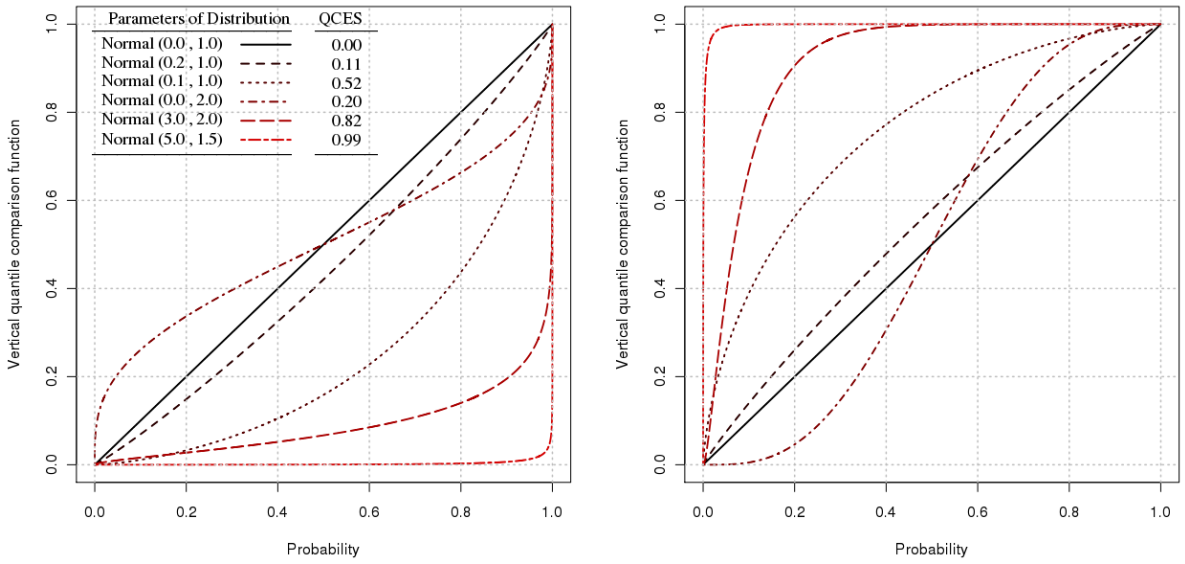
Figure 4.4: *The quantile function* (4.9) *for some Normal distributions, with associated QAD, where the Normal distribution* $N(0, 1)$ *is the baseline.*



son. We repeat this 10000 times and compute the mean and standard deviation for each ES for each simulation. All the ES increase as the mean is shifted. For the $N(0, 2)$ comparison, i.e. with the same mean and larger scale than baseline, Cohen's d and Cliff's δ ES remain close to 0 (no effect), but QCES=0.214 which would suggest a *medium* effect. This illustrates that Cohen's d and d_r are not particularly sensitive to scale changes. The unbounded measures – Cohen's d, KL, and the QAD – have larger standard deviations as the value of the ES increases. The bounded measures – Cliff's δ and the QCES – tend to have smaller standard deviations for larger ES. When one distribution is dominant, this is reflected in both Cliff's δ and the QCES having approximately similar values. As some of these ES are skewed, we also computed the medians and approximate 95% probability intervals on the median for each ES. These gave a similar interpretation.

Figure 4.4 shows the quantile function (4.9) for these distributions, with the solid line representing the quantile function of the baseline $N(0, 1)$. Consider distribution functions F and G with means μ_F, μ_G and standard deviations σ_F, σ_G. As the mean increases, the quantile function shifts upwards. In the case of $\sigma_F = \sigma_G$, this shift is the increase in the mean. The QAD effect size, $K(F\ ,\ G)$, is the area between the quantile functions for F and G. For Normal distributions with equal spread, $K(F\ ,\ G) = |\mu_F - \mu_G|$. In

Figure 4.5: *The vertical comparison quantile function* (4.10) *for some Normal distributions, with associated QCES. The left panel shows* $V_F^G(p) = G(F^{-1}(p))$. *The right panel shows the corresponding function* $V_G^F(p) = F(G^{-1}(p))$.



the case where $\mu_F = \mu_G$ and $\sigma_F < \sigma_G$ the quantile functions intersect, with the quantile function for G distorted to lower values at lower probabilities and higher values at higher probabilities, indicating heavier tails. Where $\mu_F \neq \mu_G$ and $\sigma_F \neq \sigma_G$, there is typically both shift and distortion. In contrast to ES such as Cohen's d, the QAD is sensitive to differences across the full distributions and so produces a larger value for the QAD when the standard deviation changes but the mean parameter stays fixed.

Figure 4.5 plots the vertical quantile comparison functions (4.10) between distributions F and G, using F as baseline in the left-hand panel, and – reflected – G as baseline in the right-hand panel. $F = G$ corresponds to the solid line $y = x$ in both panels. When two distributions are close, $V_F^G(p)$ and $V_G^F(p)$ are functions close to $y = x$. Specific types of departure from the solid line suggest potential orderings of the two distributions (Barlow and Proschan, 1975). Intersections indicate that the distributions may have close mean parameters and different standard deviations. A function lying strictly above or below the solid line indicates that one of the distributions stochastically dominates the other. When one distribution is very far from the other, the vertical quantile comparison function is far from the solid line. The QCES is computed using the area(s) between the curve and the solid line. Thus, the more dissimilar the distributions, the larger the QCES.

Figure 4.6 illustrates how the value of the QAD (4.11) changes as we vary the mean and

Figure 4.6: *Contour plot of the QAD* (4.11) *for Normal distribution comparisons with* $N(0,1)$ *baseline: changes in ES as we vary* μ *and* σ*. The point (0,1) locates the baseline. ES are positive unbounded.*

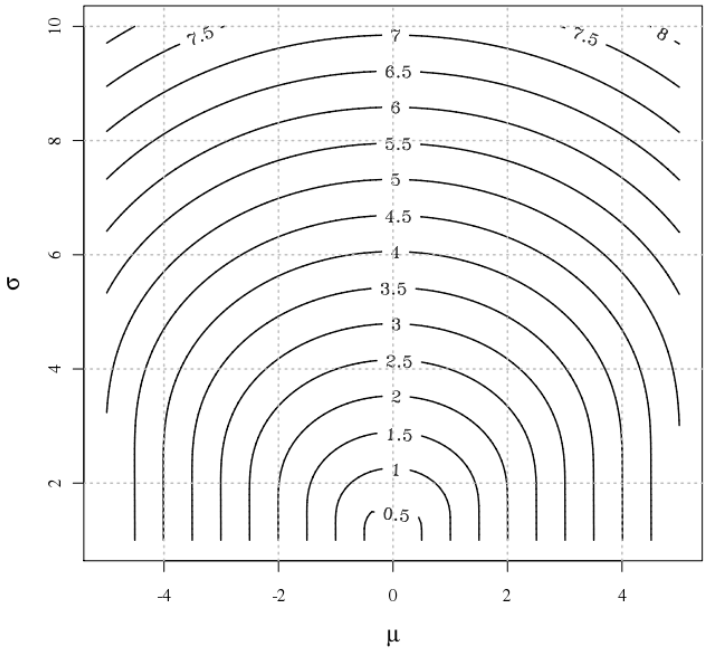
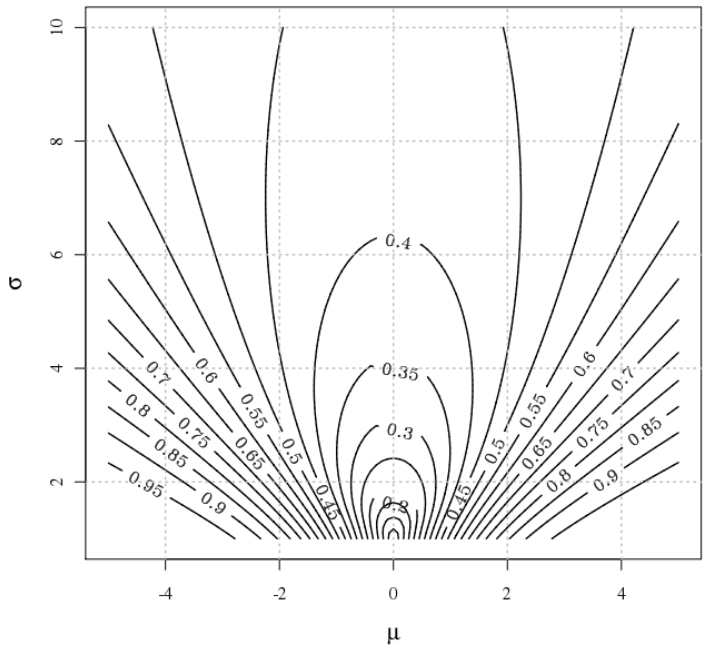


Figure 4.7: *Contour plot of the QCES* (4.14) *for Normal distribution comparisons with* $N(0,1)$ *baseline: changes in ES as we vary* μ *and* σ*. The point (0,1) locates the baseline. ES are in (0,1) by design.*



standard deviation. Large differences in mean correspond to larger ES. For fixed means, large differences in the scale parameter lead to larger ES. Differences in both location and scale lead to even larger ES. An equivalent graph for the QCES (4.14) is illustrated in Figure 4.7. This ES is always in (0,1), with higher values representing large effects. Note that because this measure highly depends on the degree of overlap of distribution functions, the ES will not change substantially when one of the distributions has a large standard deviation relative to the other.

4.5 Two-sample Weibull distribution comparisons

In this section we investigate the behaviour of these ES for the Weibull distribution (Weibull, 1951), which has been used to describe the statistical behaviour of many phenomena due to its flexibility. We have already illustrated the Weibull distribution and its properties in §3.2.2. Recall that the shifted Weibull distribution for a random variable $X \sim W(\alpha, \lambda, \theta)$ has pdf:

$$f(x; \alpha, \lambda, \theta) = \frac{\alpha}{\lambda}\left(\frac{x-\theta}{\lambda}\right)^{\alpha-1} \exp\left\{-\left(\frac{x-\theta}{\lambda}\right)^{\alpha}\right\} \qquad x \geq \theta, \tag{4.21}$$

where $\alpha > 0$ is a shape parameter, $\lambda > 0$ is a scale parameter, and θ is the threshold or location parameter. When $\theta = 0$, this function reduces to the two-parameter distribution, and we use the notation $X \sim W(\alpha, \lambda)$.

We compare Weibull distributions with different shape and scale parameters to a baseline $W(1, 1)$ distribution. This is equivalent to an exponential distribution with rate parameter 1. We choose counterpart Weibull distribution parameters in such a way so as to cover the variety of shapes the distribution may take, as shown in Figure 4.8.

As for the Normal distribution comparison, we carried out a similar Monte Carlo exploration of ES for Weibull distribution comparisons with a $W(1, 1)$ baseline. Means and standard deviations for simulated ES are given in Table 4.2. Small changes in the shape parameter produce small ES for all measures. Larger changes in shape parameter have only a small impact on Cohen's d and Cliff's δ, suggesting that these do not capture such changes well. The QCES for the $W(1.7, 1)$ and $W(0.5, 1)$ comparisons to baseline indicate that these changes in shape and scale have a larger practical effect than Cohen's d and Cliff's δ suggests.

Figure 4.9 shows the corresponding quantile functions, with the solid line representing the quantile function for the baseline $W(1, 1)$ distribution. As the scale parameter λ increases,

Table 4.2: *Mean and standard deviation of Monte Carlo simulations of Cohen's d, Cliff's δ, the QAD and QC Effect sizes, and the KL divergence, for Weibull distribution comparisons with $W(1,1)$ baseline.*

		Effect Size				
		Cohen's d	Cliff's δ	KL	QAD	QCES
$W(1.0,1)$	mean	-0.000	0.009	0.120	0.130	0.071
	sd	(0.143)	(0.082)	(0.067)	(0.079)	(0.045)
$W(1.2,1)$	mean	-0.064	0.025	0.154	0.163	0.090
	sd	(0.141)	(0.084)	(0.065)	(0.075)	(0.043)
$W(1.7,1)$	mean	-0.126	0.075	0.349	0.296	0.174
	sd	(0.137)	(0.079)	(0.069)	(0.070)	(0.040)
$W(0.5,1)$	mean	0.322	-0.087	1.185	1.231	0.217
	sd	(0.094)	(0.081)	(0.491)	(0.398)	(0.038)
$W(2.0,3)$	mean	0.953	0.239	0.880	0.878	0.600
	sd	(0.213)	(0.018)	(0.108)	(0.088)	(0.061)
$W(3.0,3)$	Mean	1.718	0.289	1.555	1.690	0.788
	sd	(0.232)	(0.002)	(0.242)	(0.130)	(0.046)

the quantile function shifts upwards. Different shape parameters lead to distorted quantile functions which intersect the quantile function for the baseline $W(1,1)$ distribution. The magnitude of the QAD effect size (4.11) is the area between two such quantile functions. In general, larger scale parameters produce larger ES. Figure 4.11 shows how the QAD changes as we vary α, λ from the baseline at (1,1). For a fixed value of α, an increase in λ results in a sharp increase in ES, and especially for $\alpha < 1$.

Figure 4.10 displays the vertical shift quantile functions and the associated QCES (4.14) for these comparisons. Changing the scale parameter affects the ES considerably; changes in shape have less impact. The contour plot shown in Figure 4.12 summarises changes in ES as we vary (α, λ) from the baseline point $(1,1)$.

4.6 Application: analysis of clickstream data

In this section we use the proposed effect sizes of the chapter for an application in clickstream data analysis. Analysis of clickstream data can help to enhances understanding and prediction of website visitor behaviour (Andersen et al., 2000), usually with the aim of

Figure 4.8: *The pdfs for the compared Weibull distributions.*

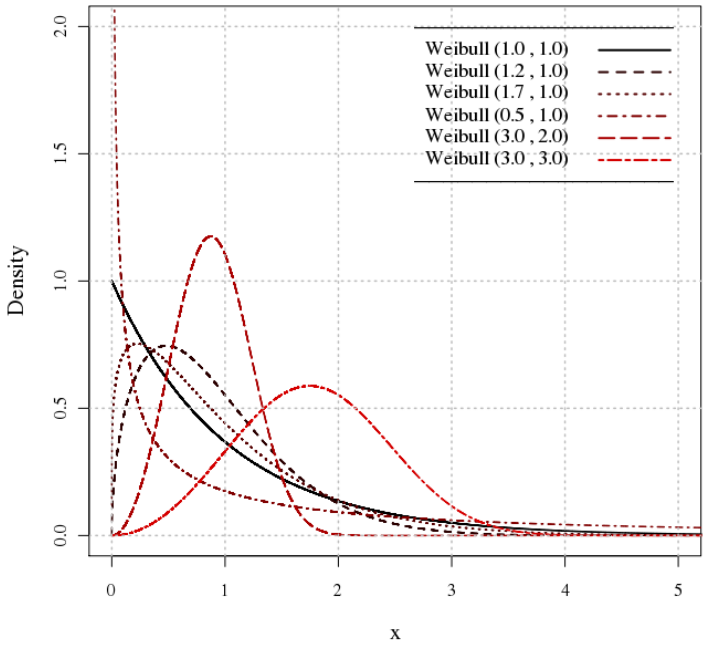


Figure 4.9: *The quantile function* (4.9) *for some Weibull distributions, with associated QAD comparing to a baseline* $W(1,1)$ *distribution.*

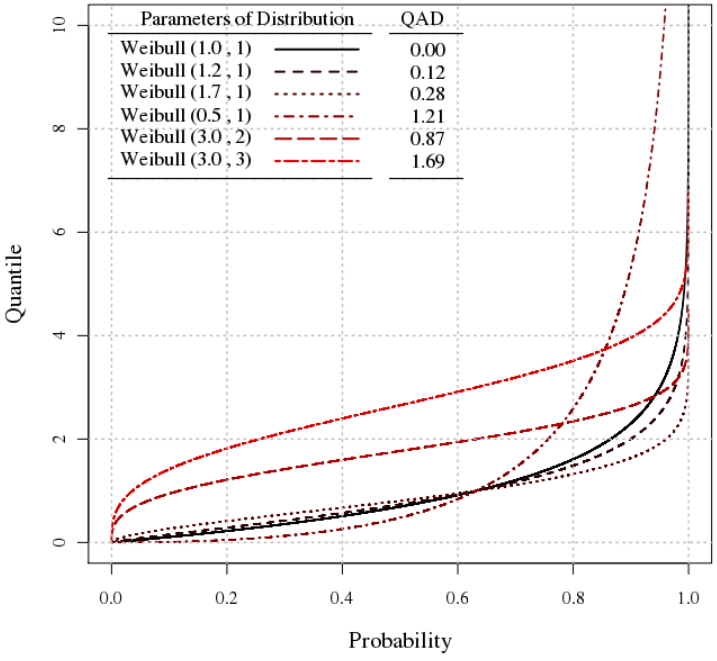


Figure 4.10: *The vertical comparison quantile function* (4.10) *for some Weibull distributions, with associated QCES comparing to a baseline* $W(1,1)$ *distribution. The left panel shows* $V_F^G(p) = G(F^{-1}(p))$*. The right panel shows the corresponding function* $V_G^F(p) = F(G^{-1}(p))$*.*

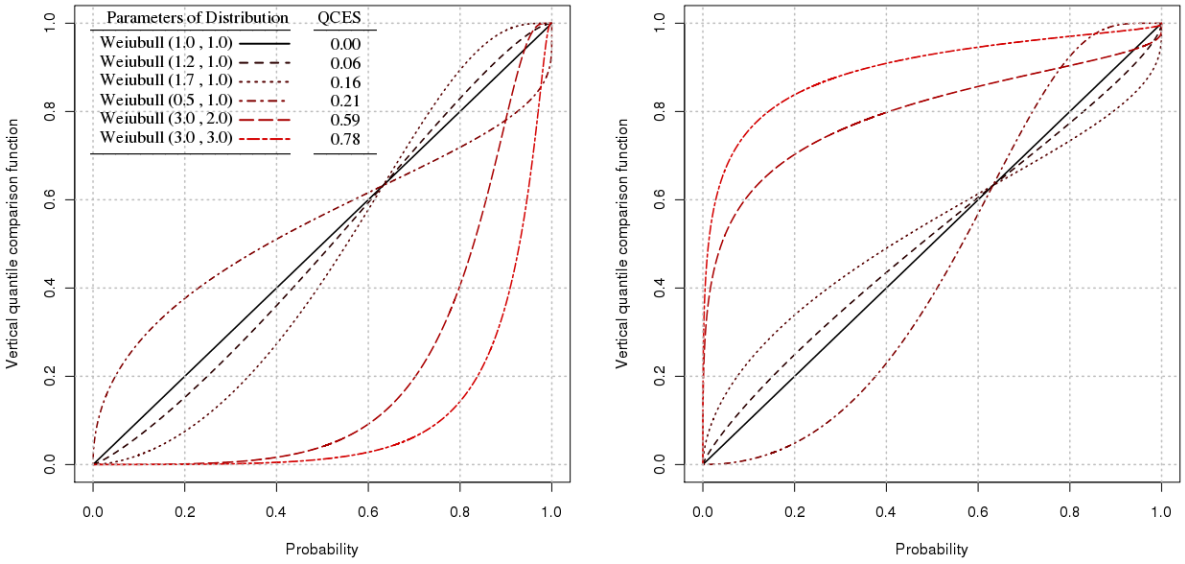


Figure 4.11: *Contour plot of the QAD* (4.11) *for Weibull distribution comparisons with* $W(1,1)$ *baseline: changes in ES as we vary* α, λ*. The point (1,1) locates the baseline. ES are positive unbounded.*

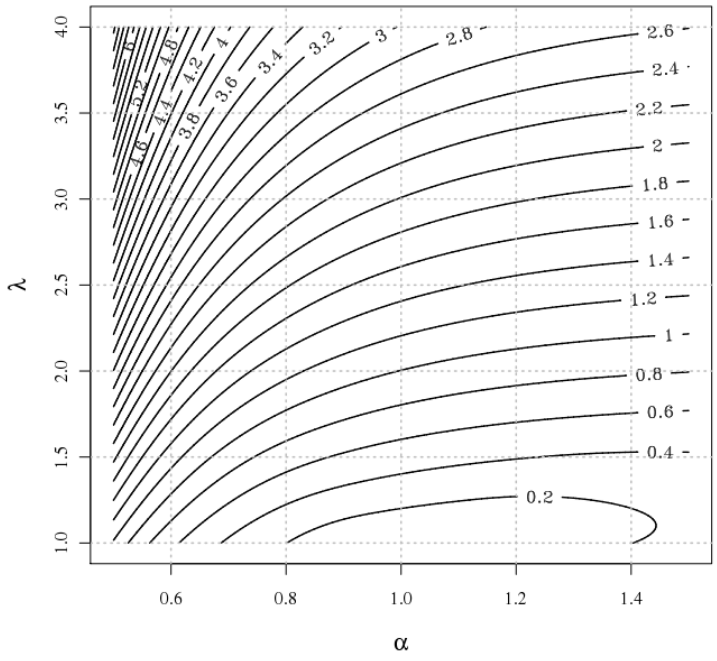


Figure 4.12: *Contour plot of the QCES* (4.14) *for Weibull distribution comparisons with* $W(1,1)$ *baseline: changes in ES as we vary* α, λ*. The point (1,1) locates the baseline. ES are in (0,1) by design.*

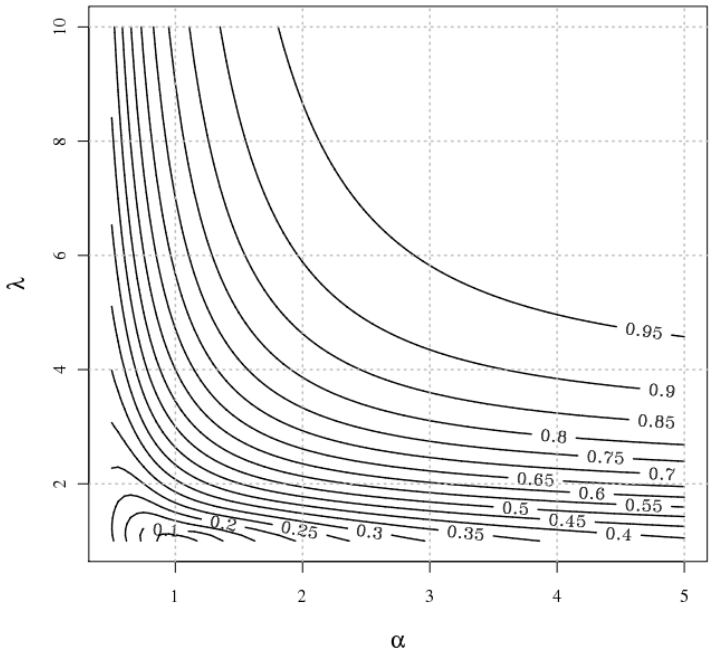
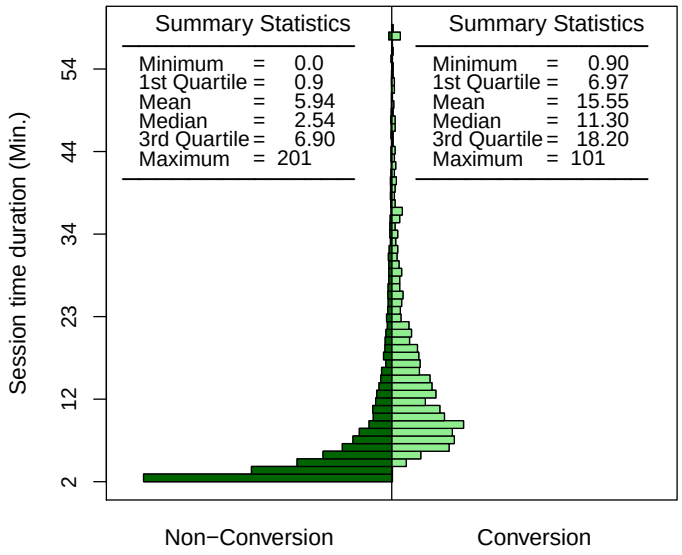


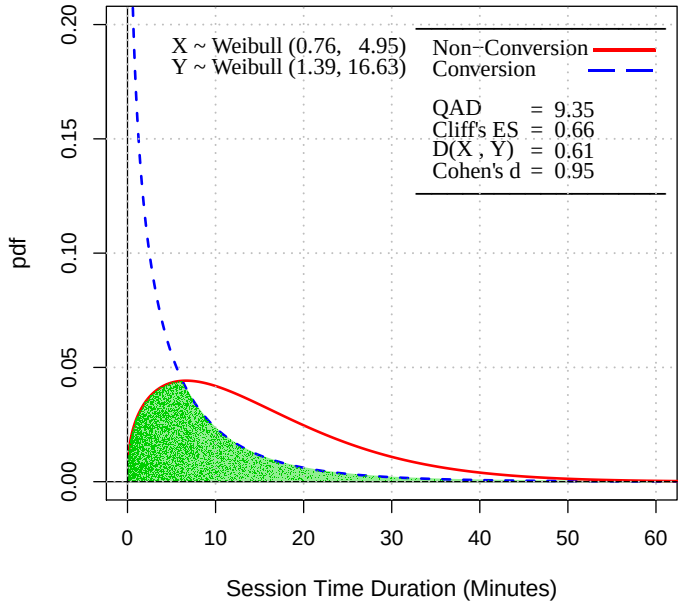
Figure 4.13: *Back-to-back histograms of session time duration for 1,353 website visits resulting in a sale, and durations for 8,747 non-sale visits.*



maximising customer sales and revenue. Exploratory analysis of such data, even though often superficial, is useful for improving system performance and providing marketing decision-support (Markov and Larose, 2007). A simple common question, for example, is whether the revenues extracted from customers arriving from different entry points is about the same or different. This is a simple two-sample comparison. The amount of data available is huge, potentially millions of records per day. Therefore, standard methods for exploring such simple hypotheses fail as they inevitably yield tiny p-values. Furthermore, the underlying populations are often highly non-Normal. We may instead employ ES to explore such hypotheses, for which we need the ES methods developed in this chapter. We use SLC data introduced in chapter 1. Recall that this clickstream data is collected for one website for one week in the summer of 2008. Some data cleaning is required in order to remove duplicates and so forth. In all, 10496 visit details were recorded.

There are very many potential relationships to explore. For this illustration, we examine session time duration, T, the time a customer spends on the website. We exclude customers who visit only one page. Figure 4.13 shows histograms of T, right censoring at 100 minutes, with visits classified as leading to sales, T_1, or no sale, T_0. The distributions

Figure 4.14: *Fitted pdfs of visit time duration T, separately for sales and non-sales visits, estimation via maximum likelihood.*

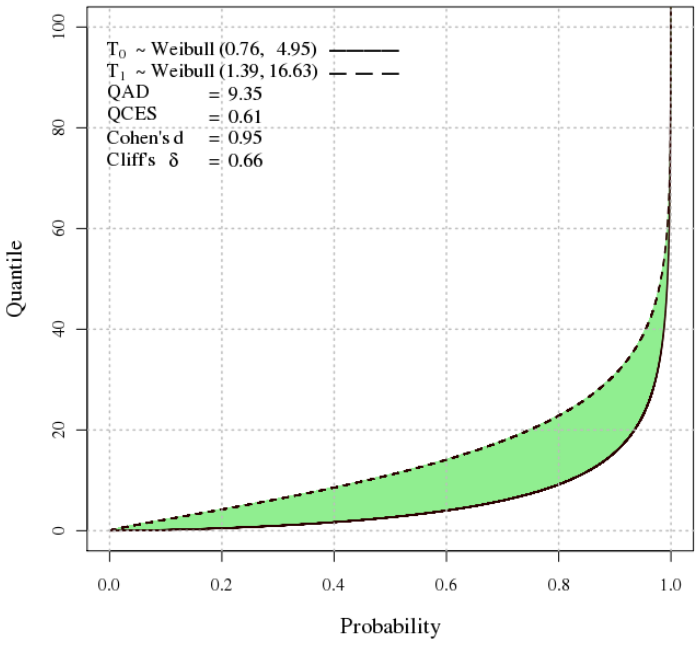


are visually quite different. Summary statistics suggest larger values of T for visits which lead to sales. In testing whether the means of the two populations T_0 and T_1 are equal, the standard t-test is not much help. The distributions seem clearly non-Normal, and the sample size is so large that tiny p-values necessarily result. The Mann-Whitney test has similar drawbacks. Cliff's nonparametric ES is $\delta = 0.22$, implying some degree of dominance. Cohen's $d = 0.95$ suggests a large effect under an assumption of Normality, but this is inappropriate here. We thus compute the quantile-based QAD and QCES effect sizes. We may compute these either by using empirical cdfs or by fitting suitable distributions to each set of observations.

4.6.1 Fitting a distribution and then computing effect sizes

We fit separate Weibull distributions to T_0 and T_1 via maximum likelihood. The fits are shown in Figure 4.14. The shaded area is the overlap in the two distributions. The shape parameters are $\alpha_1 = 1.39 > 1$ and $\alpha_0 = 0.76 < 1$ for the sales and non-sales groups respectively. The sales-group distribution also has a larger scale parameter. Quantile

Figure 4.15: *The fitted quantile functions for session time duration for visit time duration, separately for sales and non-sales visits.*



plots suggested that these fits were good except in the furthest extremities. The quantile functions for these distributions are shown in Figure 4.15. The shaded area is the QAD (4.11), which evaluates to $K(T_1\ ,\ T_0) = 9.35$, suggesting that on average the quantiles of T for the sales group exceeds quantiles in the non-sales group by 9.35 minutes. It should be noted that The QAD and QCES do tell us which distribution is dominant, when a dominance exists. At this point complimentary graphs can help to infer about ordering and effect size just shows the magnitude of effect.

Figure 4.16 depicts the vertical quantile comparison functions (4.10) for these distributions. The QCES is the area between these functions and turns out in this case to be QCES= 0.61. Following the argument in §4.3.3, we would judge this as indicating a very strong ES.

In order to assess sensitivity, we take the $W(0.76, 4.95)$ distribution for T_0 as baseline and calculate the QCES when we vary the fitted parameters of the distribution for T_1. Figure 4.17 shows the resultant contour plot. The baseline is indicated by a black square. The actual QCES is indicated by a shaded circle. The plot suggests that the sales group has fitted parameters which are very far from parameter values which would produce a much smaller ES.

Figure 4.16: *Vertical comparison quantile functions for T_0 and T_1.*

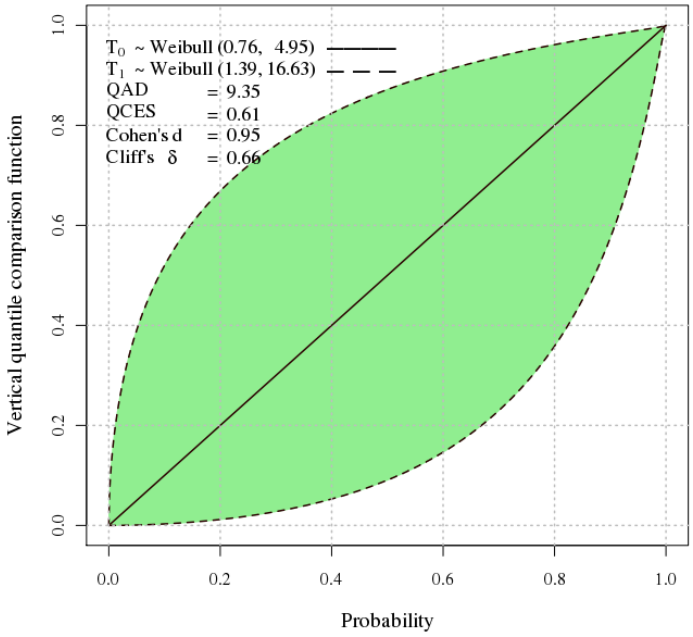
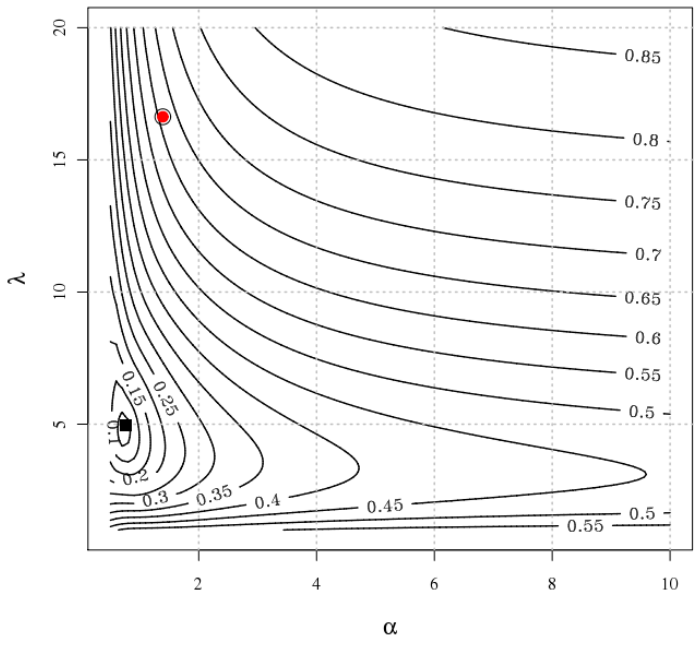


Figure 4.17: *Sensitivity plot: a contour plot of QCES values for varied parameter choices for sales-group distribution, comparing to baseline distribution T_0.*



4.6.2 Using empirical quantile functions to generate effect sizes

We may instead use the empirical quantile functions, derived from the empirical cdfs, to compute ES. This avoids the need to estimate parametric forms for the variables under study. Using the empirical cdfs, we computed QAD=9.08 which is slightly smaller than the corresponding parametric estimate. The corresponding value of the QCES is 0.654, again suggesting a very large effect size.

4.7 Inference based on bootstrapping

The theoretical distributions of the QAD and the QCES are complicated to find. Thus, we derive a bootstrap approximation of their distributions given the observed samples (Davison and Hinkley, 2006). For the two-sample comparison, repeated resamples are drawn with replacement from the two sets of observations and ES and so forth calculated from the resamples. We repeated this resampling process 10,000 times in order to find bootstrap distributions and confidence intervals for our ES (Hesterberg et al., 2003). We did this for the parametric method described in §4.6.1, which involves fitting Weibull distributions to each resample and then computing ES, and for the nonparametric method described in §4.6.2, which generates a different empirical cdf for each resample. Table 4.3 shows the result of the bootstrap sampling for QAD and QC effect sizes for both parametric and non-parametric approaches. F_0 and F_1 are the fitted cdfs for T_0 and T_1 respectively, and $\widetilde{F}_0$ and $\widetilde{F}_1$ are the corresponding empirical cdfs. The column entitled *Observed* gives the ES computed from the original sample. The remaining columns summarise the bootstrapped ES. Histograms of the bootstrapped ES are all reasonably Normal in shape, and the summaries give no cause for concern.

Table 4.3: *Bootstrap summry statistics based on 10,000 resamples: estimation of the standard error, 95% confidence interval, and bias for each ES.*

	Effect Size	Observed	Mean	SE	95% CI	Bias
Parametric	QAD $K(F_0\,,\,F_1)$	9.346	9.328	0.346	(8.65 , 9.99)	+0.018
Empirical	QAD $K(\widetilde{F}_0\,,\,\widetilde{F}_1)$	9.081	9.056	0.347	(8.39 , 9.74)	+0.025
Parametric	QCES $D(F_0\,,\,F_1)$	0.611	0.610	0.009	(0.59 , 0.63)	+0.001
Empirical	QCES $D(\widetilde{F}_0\,,\,\widetilde{F}_1)$	0.645	0.646	0.009	(0.63 , 0.66)	−0.001

4.8 Discussion

The standard ES proposed for two-sample comparison studies are mostly based on the assumption of the Normal distribution and limited aspects of changes in parameter. In this chapter we have introduced two ES measures, the QAD and the QCES, which are based on quantile functions and which better summarises differences distributions over the full range of probabilities. The QAD is an ES for which differences between distributions are summarised in terms of the original units of measurement. The QCES has been developed as bounded, standardized, divergence measure for circumstances where the unit of measurement is not meaningful or relevant. We have investigated these ES for two parametric families and in a practical application, and suggested some practical thresholds for what constitutes small and large effects.

Chapter 5

Conversion Analysis: Logit Model

In this chapter we use the logit model to assess the contribution of predictors to purchasing behaviour for the SLC data set. This model helps to predict the probability of an online purchase during a website visit session. A set of session attributes obtained by clickstream data are used as predictors. The subset of variables, including interaction terms, is selected from the session attributes using stepwise variable-selection techniques. This enables us to derive individual purchase probabilities for each visit based on general clickstream behaviour. It should be noted that we use just the general clickstream data of users navigating the website, whereas in the case of using registered customers one may access information about the users in more detail (for example, history of customers purchases and customer demographics). Having the best model fitted, the predictive power of the model for classifying customers concerning their purchase behaviour on the Internet is assessed. In comparison with previous studies, our contribution is to take into account interaction terms, as well as main effect general clickstream information. We intend to identify the most significant predictors of online purchasing which maximize the predictive power of our model in practice. This chapter contains initial exploratory modelling and graphing to illustrate some of the concepts and introduce some of the key issues. We do not perform exhaustive modelling of all relationship as this is not the focus of this thesis.

5.1 Introduction

A well-known feature of online shopping is that visitors of e-commerce websites are rarely loyal to a specific website when searching for a particular product/service or category (Van den Poel and Buckinx, 2005). Visitors can search several e-commerce websites for

a product/service in a relatively short time at no cost and select the one which satisfies them the best. Thus, clients are able to easily compare the offers of several companies in an e-business environment. This also leads companies to a more intensive competition arena. On the other hand, every year the number of people who do online shopping increases and people do not look at internet only as a source of information. Therefore, many companies show interest in running an active and successful e-business along with their offline media. For this reason, successful e-marketing may require a special effort to improve knowledge about visitors in order to capture a reasonable share of the market.

Clickstream data obtained by online virtual store can be used as a source of information with regards to the customers buying behaviour. Clickstream data typically contains information about the behaviour of visitors to the company's website. Investigating the behaviour of visitors to the company's website, online buyers and non-buyers, may provide a better understanding of the characteristics of visits with respect to shopping. An important metric which is computed using clickstream data is the conversion rate of a website, defined as the percentage of website visits that lead to a purchase, is of great importance for an e-commerce website manager. Consequently, there has been substantial interest in analysis of conversion using clickstream data. This analysis may provide a better perception of online buying behaviour, as it helps to improve the conversion rate by examining the motives for purchases (Sismeiro and Bucklin, 2004).

Moe and Fader (2000) was an early effort to investigate customer conversion rates, which introduced a dynamic model using behavioural changes over time to forecast internet behaviour. Later, they extended the application to historical visiting data and the type of customer visit (Moe and Fader, 2001). Padmanabhan et al. (2001) introduced a model to predict the probability of a purchase for the remainder of a session. Sismeiro and Bucklin (2004) show that experience as well as browsing behaviour can improve prediction of online buying. Van den Poel and Buckinx (2005) applied a logistic regression model for predicting online-purchasing behaviour using web browsing data as well as customers demographic data and purchase records. They used a set of variables as a detailed clickstream data, such as number of pages visited related to products, supply procedures, and personal information.

5.2 Response and explanatory variables

The SLC data, introduced in chapter 2, is used to fit the logit model on conversion status based on general clickstream information available in the SLC data. First, we randomly

split the data set into two equal parts: *training data*, to fit the model and estimate the parameters of the model and *test data* to assess the ability of the model for classification purposes.

The response variable is conversion status (CVS). The CVS, as a binary response variable, indicates whether an online purchase has been made through a web session. We assign the value of 1 for the session in which an online purchase occurs, and 0 otherwise. As reported in chapter 3, the conversion rate is 13.0%, which is hight level for an e-commerce website.

The explanatory variable most often used in the literature is the frequency of visits. Moe and Fader (2000) shows that the conversion rate is significantly higher for visitors who often return to the website. They also investigate the effect of the recency variable in a logistic regression as additional information to consider customers' behaviour. The accumulated visits variable has also proved to be a powerful indicator of purchase potential (Moe and Fader, 2001). Padmanabhan et al. (2001) found that the length of time a visitor spends on a website is positively associated with potential purchase. Bucklin et al. (2002) took into account the cumulative number of pages viewed as depth of visit, and showed that it affected the propensity to continue browsing along with measures of repeat visits. For a complete list of clickstream attributes used for modelling conversion see Van den Poel and Buckinx (2005). For the list of explanatory variables in SLC data used in the model selection see Table 5.1.

5.3 Explanatory analysis of conversion

In this section we use graphical representations to show how general clickstream data can explain the browsing behaviour in the session in terms of conversion rate. Specifically, we explore the conversion in a session for SLC data by investigating session attributes separately for sessions with and without conversion.

In the offline situation, people tend to shop more on weekends than during the week. Now the question is whether there is evidence to show that the same pattern of shopping from e-retailers on weekends holds. Figure 5.1 (left) exhibits a bar chart of the conversion rate for different days of the week. It seems that the conversion rate is slightly higher for Sundays. The width of the bars is proportional to the traffic of the website for each day. The equal width of the bars shows that nearly the same number of visitors visits the website on different week days.

Table 5.1: The label and short description of the variables in SLC data used in the model selection. It also represent the number of levels. For continuous variables number of levels is reported 1

Variable	**Label**	**No. of levels**
logTD	logarithm of time Duration (log-Seconds)	1
logNPV	logarithm Number of pages visited	1
logFTD	logarithm of former time duration (log-Seconds)	1
NFV	Number of former visits	1
UK	Whether come from UK	2
GA	Whether come from Google Ad	2
RV	Whether repeat visit	2
VW	Visit on weekend	2
VWD	Day of the week	7
HD	Hour of the day	24
TNPV	Total number of Pages visited	1
MPD	Mean of the time page durations (Seconds)	1
StdPF	Std of the time page durations (Seconds)	1

Figure 5.1: *The bar chart represents the rate of online purchase given the frequency of return to the website. The width of the bars is proportionate to the number of observations occurring in each category.*

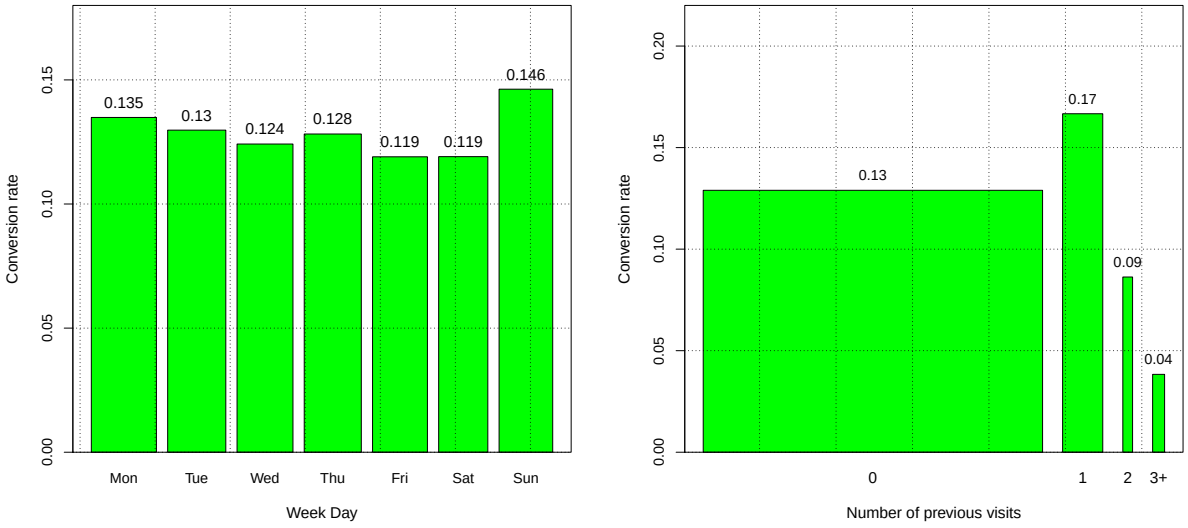
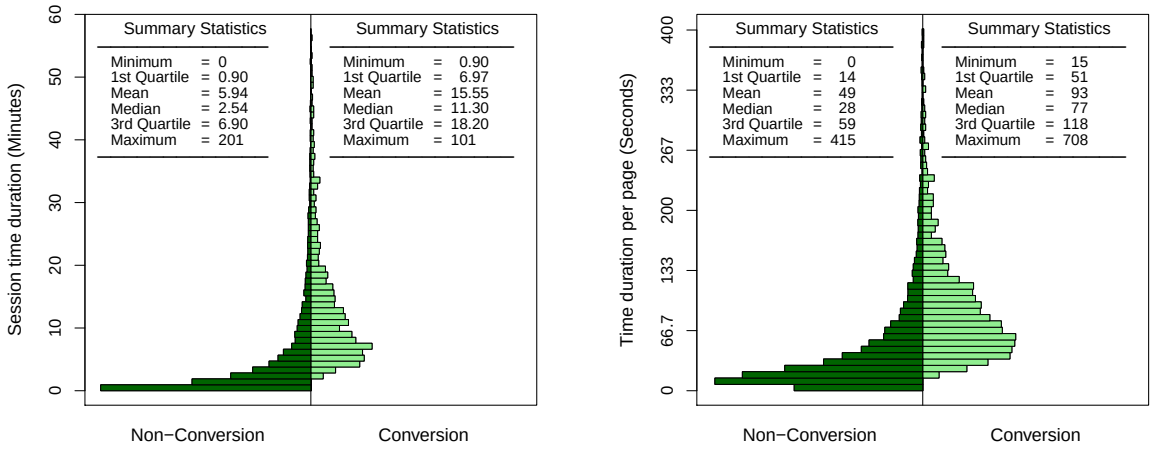


Figure 5.2: *Back-to-back histograms of session time duration (left) and average time duration per page (right) given the response variable conversion and non-conversion visits.*

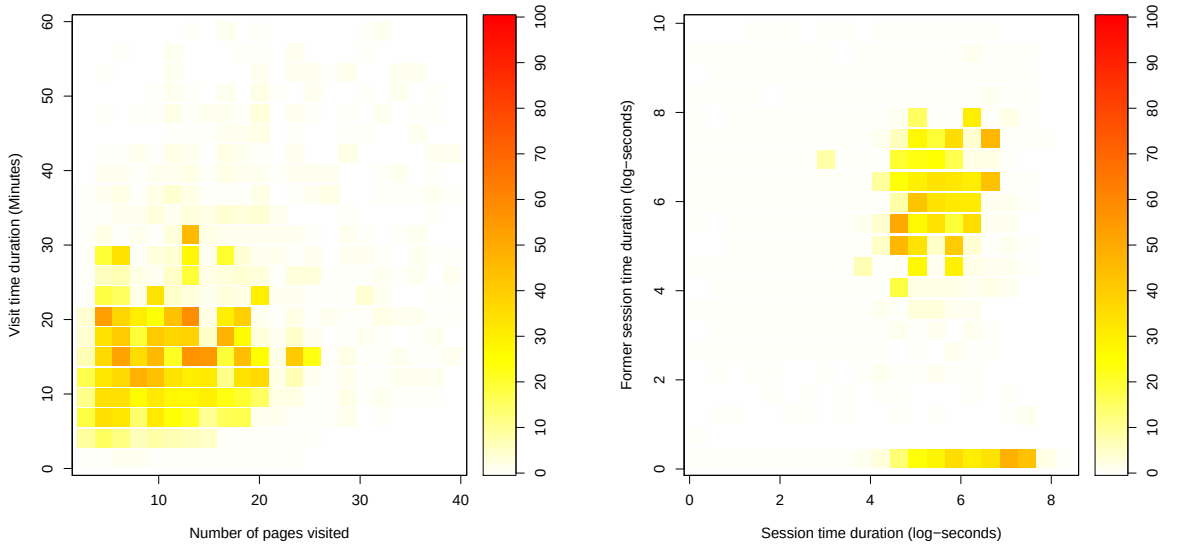


The effect of the number of times a visitor returns to the website on the conversion rate is investigated in Figure 5.1 (right). The highest conversion rate is observed for the category of one previous session. The conversion rate seems to decrease as the visitor returns to the website on several occasions (in the period of one week). It should be noted that the width of the bars is proportional to the traffic observed in the corresponding category.

The length of a session may provide valuable information about browsing behaviour. In general, it is likely for users who aim to buy to spend more time on the content web pages. There might be an exception for those who know the website very well and regularly order an item from the website. In this case, there is the need for other auxiliary information to distinguish them, such as whether the visit is a repeat sessions or total amount of time spent on website.

Figure 5.2 (left) shows the back-to-back histogram of the session time duration based on conversion segmentation. The length of a visit is visibly longer for the conversion sessions. The plot, in fact, represents the conditional distribution of the session time duration given the session with and without an online purchase. The graph also shows that in 50% of the non-conversion sessions, the visitor leaves the website within the first 2 minutes and 30 seconds of visiting it. However, only 1% of the conversion sessions last 2.5 minutes or less. Hence, increase of session time duration is associated with an increase in the probability of making an online purchase.

Figure 5.3: *Heat plot to represent the interaction effect of logarithm of session time duration, logTD, and logarithm of number of pages visited, logNPV, (left) and previous session time duration and logTD (right) on conversion rate. The heat spectrum shows the magnitude of conversion rate.*



The average time spent on each page arguably can be considered as a measure of attention to the web pages while a user is surfing a website. It is expected that more time will be spent on pages, specifically content pages, when one aims to make a purchase. Figure 5.2 (right) indicates how the distribution of the average time duration over pages differs for conversion and non-conversion sessions. The summary statistics have been computed for both groups, indicating that the median of average time per page is 77 seconds which is considerably higher than the 28 seconds computed for the non-conversion session. It is apparent that average time duration per page for conversion sessions is greater than for non-conversion sessions.

Figure 5.3 (left) portrays the effect of session time duration and number of pages viewed on purchase rate simultaneously. The colour of each square segment shows the level of conversion rate. It can be seen that, regardless of the number of pages, conversion occurs at the same rate at different levels of session time duration.

Similarly, the interaction effect of session time duration and previous session time durations on purchase rate is depicted in Figure 5.3 (right). It explains that users who make purchases in a fairly short time in the current session have previously spent a reasonable amount of time on the website. We also observe that the highest rate of purchase occurs for those who have already spent between 10 to 30 minutes visiting. This rate sharply

Table 5.2: *The conversion rate for the UK and non-UK visitors who arrived to the website through Google link or other ways. Margins give the percentages of online purchase for UK/non-UK visitors, and Google/non-Google referring to the website.*

	Non-Google	**Google**	**Total**
Non-UK	2.6 %	9.0 %	3.1 %
UK	15.2 %	12.4 %	14.5 %
Total	13.1 %	12.2 %	

decreases after 30 minutes for first-time visits, whereas for repeat-visit sessions it drops after 20 minutes.

Table 5.2 shows that domestic visitors of the website (connection from UK ISP) are less likely to engage in purchasing when coming to the website via Google Ad links., while this pattern is reversed for non-UK visitors. This may imply that the interaction term in the model on conversion is necessary. Marginal percentages show that people from the UK, in general, are more likely to purchase from the website. The conversion rate for subgroups of Google/non-Google referring does not show a difference as large as UK/Non-UK, but 0.9% increase of conversion rate in practice may be significant. Specifically, in the subgroup of UK visitors the conversion rate is 2.8% higher than Google search. It might be due to the customers who come to the website for a repeat visit directly either by typing in the address in the navigator toolbar, or using the bookmarking facility of the web navigators.

5.4 Model Specification and Estimation

The logit model is well-understood and commonly used in statistics, machine learning and many other disciplines. Its statistical foundation helps in the investigation of the relationship between discrete responses and a set of explanatory variables by means of a probabilistic model. It is a good candidate to model binary and ordinal responses, which arise in many fields of study. Several main textbooks discuss the logit models, such as Collett (1991), Agresti (1990), Cox and Snell (1989), and Hosmer and Lemeshow (1989).

5.4.1 Binary Logit Model

In the binary logit model, the response variable Y_i of the i-th individual or an experiment unit can take one of two possible values, denoted for convenience by 0 and 1 (for example, $Y_i = 1$ if an online purchase takes place during a visit session i, otherwise $Y_i = 0$). Suppose $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ is a vector of explanatory variables and $\pi_i = \Pr(Y_i = 1 \mid \boldsymbol{x}_i)$ is the response probability to be modelled. The linear logistic model is in the form of

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}, \tag{5.1}$$

where the β_js are the parameters of the model, and x_{ij} is the realization of the j-th explanatory variable for the i-th unit. The model can be presented by matrix notation: $\eta(\boldsymbol{p}) = \mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)'$ is vector of parameters, $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$, and $\mathbf{X}$ is design matrix, and $\eta(.)$ is called *link function*.

The likelihood is a function of the data and the parameters of the model. The maximum likelihood (ML) estimator of the parameters is computed by maximising the likelihood function. The form of the likelihood for the binary responses is given as:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{N} p_i^{Y_i}(1-p_i)^{1-Y_i}. \tag{5.2}$$

Two iterative maximum likelihood algorithms have been developed to estimate the ML estimate of parameters: the Fisher-scoring method, which is equivalent to fitting the model by iteratively re-weighted least squares method, and the Newton-Raphson method. Both algorithms provide the same parameter estimates; however, the estimated covariance matrix of the parameter estimators may differ slightly, as the Fisher-scoring method is based on the expected information matrix, while the Newton-Raphson method uses the observed information matrix. It should be noted that for a binary logit model the observed and expected information matrices are identical. This results in the same estimated covariance matrices for both algorithms. For details of the algorithms see McCullagh and Nelder (1989) or Hosmer and Lemeshow (2000).

5.4.2 Single Logistic Regression Models

The simplest type of logistic regression model involves only one explanatory variable; as a starting point, we first fit simple logistic regression on conversion status for each

Table 5.3: The summary of the single logistic regression model, and the labels of general clickstream measures studied

Variable	Label	df	AIC	P-Value	C-Index
↗ logTD	log Time Duration	1	6434	0.000∗∗∗	0.75∗∗
↗ logNPV	log Number of pages visited	1	7860	0.000∗∗∗	0.65∗
↗ logFTD	log of Former Time Duration	1	8075	0.494	0.51
↘ NFV	Number of Fomer visits	1	8044	0.000∗∗∗	0.52
↗ UK	Whether come from UK	1	7870	0.000∗∗∗	0.64∗
≈ GA	Whether come from Google Ad	1	8074	0.330	0.50
≈ RV	Whether Repeat Visit	1	8074	0.278	0.52
≈ VW	Visit on Weekend	1	8074	0.771	0.50
≈ VWD	Visit on Week day	6	8070	0.323	0.50
↗↘ HD	Hour of the Day	23	8050	0.000∗∗∗	0.61∗
↘ TNPV	Total Number of Pages Visited	1	8043	0.000∗∗∗	0.71∗∗
↗ MPD	Mean of the page durations	1	8013	0.000∗∗∗	0.73∗∗
≈ StdPF	Std of the Page Duratios	1	8075	0.491	0.50

† Significant at level: 0.1 (∗) 0.05 (∗∗) < 0.05 (∗∗∗)
‡ Effect size: small (∗) medium (∗∗) large (∗∗∗)

explanatory variable in the SLC data set. This helps to obtain a rough idea of the variables which have a strong association with the conversion status.

Table 5.3 shows the list of explanatory variables. Each row corresponds to the simple logistic regression fitted for a single explanatory variable. The columns of degrees of freedom and p-values correspond to the analysis of deviance, using the chi-square statistics indicating the significance of reduction in deviance. The table also shows the Akaike Information Criteria (AIC) of the model, as a penalised lack-of-fit measure, and the C-index, the area under the ROC curve, is an effect size measure showing the predictive power of the model. It should be noted that non-binary categorical variables (e.g., the effect of the weekdays on purchase) do not have one coefficient but rather have as many coefficients as there are categories minus one. Of course, not all of these variables will be in the final fitted model, but this will be an starting point to find the association of covariates to the response variable and model building.

There are seven variables with small p-values (≤ 0.000) when fitting simple linear regression, distinguished by $*$ in Table 5.3. The direction of association is also represented using the symbols: $\nearrow$ for positive, $\searrow$ for negative, and $\approx$ no association. Among these variables the NFV has a very small C-Index, 0.52. It should be remembered that a model that randomly classifies the response variable will have an average ROC area of 0.5; As a rule of thumb an C-Index of 0.65 or above is deemed practically significant in many contexts (Van den Poel and Buckinx, 2005). The strongest (positive) association to conversion belongs to the logarithm of the session time duration, logTD. The positive association for UK shows that visitors from UK are significantly more likely to make a purchase. Having positive association for MPD shows that depth of visit increase the chance of conversion. Although it is expected, the model shows that the conversion rate varies for different hours in a day. It should be noted that just because a particular explanatory variable alone does not result in a strong model that does not mean that it will not be useful when combined with other variables.

5.5 Model Selection Procedures

Following the parsimonious principle in statistics, we seek a relatively small subset of predictors which can reasonably explain the response variable. When the number of predictors increases, the number of possible interaction goes up considerably. Therefore, one needs some special algorithms for the purpose of choosing the best subset. For example, for 15 available main effect terms, there exist 105 two-way interaction terms. We

Table 5.4: *The summary of stepwise model path and the deviance analysis.*

Step	Variable	df	Res. df	RDev	AIC	BIC	P-Value
1		1	10495	8071.09	8073.09	8080.35	
2	+ logTD	1	10494	6429.73	6433.73	6448.25	<1.0e-16
3	+ UK	1	10493	6273.31	6279.32	6301.09	<1.0e-16
4	+ NFV	1	10492	6199.70	6207.71	6236.74	<1.0e-16
5	+ logFTD	1	10491	6187.33	6197.34	6233.62	4.3e-04
6	+ logTD:logFTD	1	10490	6173.63	6185.64	6229.18	2.1e-04
7	+ GA	1	10489	6164.10	6178.11	6228.91	2.0e-04
8	+ UK:GA	1	10488	6149.37	6165.37	6223.44	1.2e-04

examine the result of three widely used variable-selection algorithms: forward selection; backward elimination; and forward/backward procedures.

The forward selection method first estimates parameters for effects forced into the model. These effects are the intercepts and the effects of the first n explanatory variables we are interested in keeping. The procedure continues by computing and examining independent variables one-by-one to the logistic regression model and choosing the largest of these statistics. If it is significant at the α_{in} level, the corresponding effect is added to the model. It should be noted that, once an effect is entered into the model, it is never removed from it. The process is repeated until none of the remaining effects meet the specified level for entry. It may also be asked to terminate when a stopping value is reached.

The first step for a backward selection method includes fitting a model with all the effects we are interested in examining. At each step an effect is removed if it does not meet the significance level of stay α_{out}, or we decide to keep n specified variables in the model. The process continues until no other effect in the model can be removed or until the stopping value is reached.

The forward/backward selection is similar to the forward selection except that effects already in the model do not necessarily remain. Effects are entered into and removed from the model in such a way that each forward selection step may be followed by one or more backward elimination steps. The stepwise selection process terminates if no further effect can be added to the model or if the effect just entered into the model is the only effect removed in the subsequent backward elimination.

In addition to the variables listed in Table 5.3 we also allows the quadratic term of the continuous variables to be selected in the stepwise selection process. All three methods suggest the same model, so we have not shown the details of the forward selection and backward elimination processes. The summary of the forward/backward selection method is shown in Table 5.4, allowing all first-order and interaction terms. In the first step, the intercept-only model is fitted and individual score statistics for the potential variables are evaluated. Due to the large number of observations in our study, the Bayesian Information criterion (BIC), which will be discussed in the next section, is applied as a score statistic in the stepwise variable selection. Each addition or deletion of a variable to or from a model is listed as a separate step, and at each step the new model is fitted. At step 2, the variable logTD is selected into the model since it is the most significant variable among those to be chosen (reduction of 1632.10 in BIC), emphasizing the importance of the visit time duration in distinguishing between conversion sessions and non-conversion sessions. The UK and NFV, as well as the logFTD variables, are the three variables that explain the online purchase. Through 8 steps, the variable selection methods result in the contribution of 4 main effects and 2 two-way interaction effects. Finally, in step 9, the remaining variables outside the model do not meet the entry criterion, and the forward/backward selection is terminated.

It should be noted that using AIC as score statistics in the forward/backward selection results in a model containing 6 extra interaction terms. The contribution of these interaction effects is not significant and the large number of observations is the main reason for illusory merit of entrance. Considering the single logistic model fitted in Table 5.3, MPD have not allowed to get in the model. This can be explained by the multicollinearity between the variables in the model and these two variable. For example, there is positive correlation between logTD and MPD.

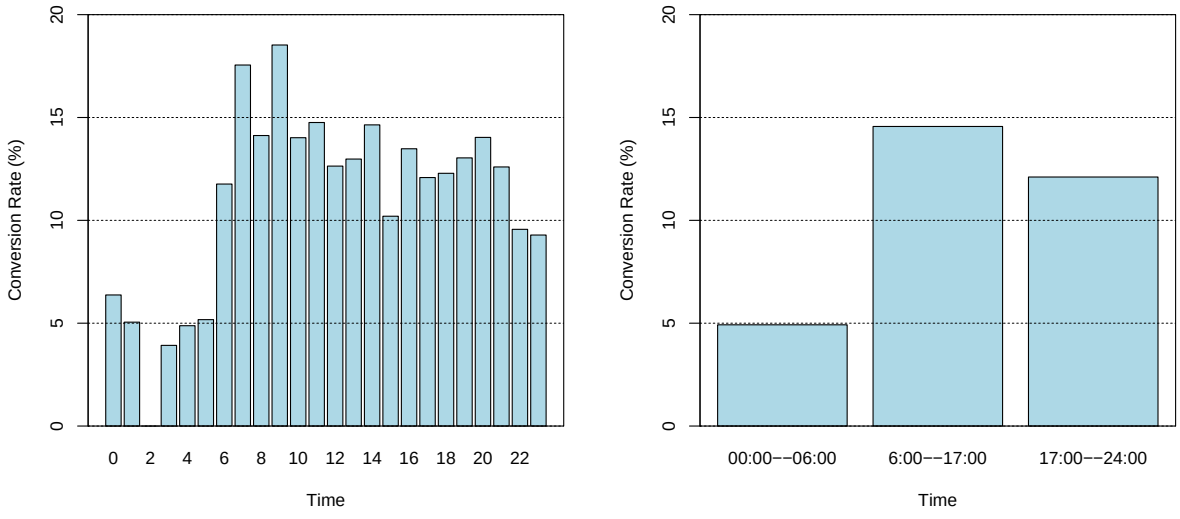
The NPV variable does not meet the criterion to enter to the model, but that does not necessarily mean that there is no relationship between purchase and number of pages visited. This might be due to the high correlation of variable logNPV and logTD (0.66 in our sample) that causes the model to have both effects simultaneously. As the VWD do not meet the entry criterion of the model, there is no evidence showing that people tend to purchase from an e-retailer at the weekend or on any particular day of the week more than any other. We observed in Figure 5.1 that the conversion rate is slightly higher for Sundays. However, this does not cause a considerable reduction in model predictive quality and its effect is not in the final selected model.

The hour of the day has been found to be a highly influential factor, as shown in Table 5.1 which consider both p-value and C-Index. This factor was not selected in the stepwise

Table 5.5: *The summary of stepwise model path and the deviance analysis.*

Step	Variable	df	Res. df	Dev	AIC	BIC	P-Value
1		1	10495	8071.09	8080.35	8089.60	
2	+ LogTD	1	10494	6429.73	6448.25	6457.49	<1.0e-16
3	+ UK	1	10493	6273.32	6301.10	6310.34	<1.0e-16
4	+ NFV	1	10492	6199.70	6236.74	6245.97	<1.0e-16
5	+ Hour.C2	1	10491	6174.37	6220.66	6229.90	<1.0e-04
6	+ logFTD	1	10491	6162.22	6217.77	6227.00	<1.0e-04
7	+ LogTD:logFTD	1	10490	6148.46	6213.27	6222.50	<1.0e-04
8	+ GA	1	10489	6138.29	6212.36	6221.58	<1.0e-04
9	+ UK:GA	1	10488	6123.56	6206.89	6216.11	<2.1e-04

Figure 5.4: *Bar-plot for the the conversion rate at different hours of the day (left) a three period of the the day (right)*



approach. It may be because this factor imposes 23 degrees of freedom. Figure 5.4 (left) shows the conversion rate for different hours of the day. We also show the rate for *over night, day time* and *evening*, Figure 5.4 (right). The graph shows that the conversion rate over night is nearly half that of the rest of the day. We recode this factor more parsimoniously into a three-level, as shown on the Figure 5.4 (left) and also a two-level time period, considering 00:00–06:00 versus the rest of the day. We refer those recoded factors as Hour.C3 and Hour.C2 respectively. It is quite obvious that the conversion rate during the day, 6:00–17:00, is three times more than night time, 00:00–06:00. This difference is much less noticeable when comparing the day time versus evening time. Performing the forward-backward selection using both scenarios showed that only the two-level time category, Hour.C2, is significant enough to appeared in the final model. The summary of the forward/backward selection method is shown in Table 5.5. The result shows that in the fifth step the Hour.C2 is entered to the model. All factors selected in Table 5.4 still remain in the final model.

5.6 Parameter Estimation and Visualisation

Unlike linear regressions, whose coefficients provide a direct multiplicative interpretation, for logistic regressions it is more difficult (but still possible) to directly interpret the magnitude of a coefficient. Thus, one may simply interpret the sign as indicating a positive or negative effect and try to indicate magnitude using other means like building contingency tables and auxiliary graphs. It is also more complicated to interpret the coefficients of explanatory variables of the model when the corresponding interaction terms are significant. We will take advantage of data visualization techniques to assess the interaction effects.

Table 5.6 lists the parameter estimates, their standard errors, z-score statistics for individual parameters, and corresponding p-values. It shows that session time duration is the most significant effect of the model. It is also supported by the back-to-back histogram depicted in Figure 5.2. Due to presence of interaction terms in the model, the coefficients need to be interpreted carefully.

In order to clarify how interaction term affects the probability of the conversion, we applied an interaction plot for one continuous effect and also for the case of two categorical effects. Figure 5.5 (left) shows two separate logistic curves, for session which arrive at the website through Google and non-Google arrivals, based on the variable (log) session time duration. It should be noted that the corresponding interaction term, GA:logTD, is not

Table 5.6: The maximum likelihood estimate of the parameters of the logit model and corresponding test statistics

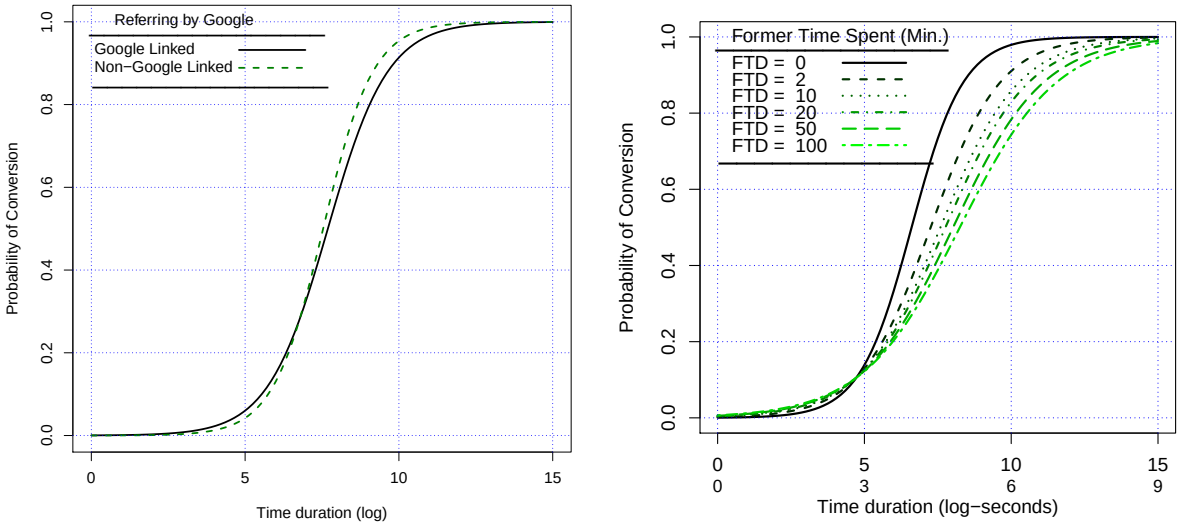
Coefficient	Estimate	Std Error	Z-value	P-Value
(Intercept)	-11.187	0.431	-25.95	< 2.0e-16
logTD	1.135	0.030	37.53	< 2.0e-16
UK	1.848	0.145	12.75	< 2.0e-16
NFV	-0.534	0.094	-5.68	1.4e-08
Hour.C2	1.034	0.185	5.56	2.7e-08
logFTD	0.377	0.064	5.91	3.5e-09
GA	1.457	0.329	4.43	9.6e-06
logTD:logFTD	-0.052	0.011	-4.65	3.3e-06
UK:GA	-1.767	0.336	-5.25	1.5e-07

significant in the model. The slight difference of two lines emphasises the negligible effect of interaction term in predicting purchase behaviour in the model. One can also infer from the Figure 5.5 (left) that probability of conversion slightly increases up to 150 seconds (2 minutes and 30 seconds corresponds to the logarithm of 5) of the session duration, and then it shows a sharp increase.

Figure 5.5 (right) provides a graphical representation of interaction effect of logarithm of time spent on the website in previous visits (logFTD) and the logarithm of the current session time duration (logTD) in the logistic regression on conversion. This interaction term has been found statistically significant in the final model. The plot shows considerable distance between curves for different values of former time spent on the website. We observe that visitors who spend less than 2 minutes and 30 seconds on the website (i.e. logTD less than 5) are more likely to purchase if they have visited the website before, but this difference is not considerable. On the other hand, for those spending more than 5 minutes visiting, the increase in the probability of purchase rises more sharply for first-visit surfers (solid line or FTD= 0), and gradually decreases as the previously-spent time increases. This plot also let us compare the probability of conversion based on interaction effect model. For example, the logTD= 7 and FTD= 0 the probability of conversion is estimated to be 0.6, whilst this probability decrease to 0.4 for FTD= 10.

The frequency of return to the website has been found significant in the model. The negative sign of the parameter estimate describes that several return visits to the website would decrease the likelihood of purchase in the session. However, because of the presence of the interaction and quadratic terms in the interpretation, the main effects need to be

Figure 5.5: *Interaction plot for the logistic regression, the effect of logTD×GA (left) and the effect of logTD×logFTD (right).*



interpreted carefully.

5.7 Assessing Model Quality

Once we have created a model, we need to assess how *good* it is, both in absolute terms and when it is compared with competing models. There are several classes of model quality criteria.

Residual Deviance: The statistic $-2\times$Log-likelihood, sometimes referred to as the residual deviance (RDev), is a measure of how far a particular model is from the ideal model that perfectly fits the training data set, where the ideal deviance is considered to be 0. The deviance for only-intercept model which is called the null deviance is computed at the first step. The null deviance is actually the measure of the worst-possible model for predicting a given response variable, since it does not take any explanatory variables into account.

The residual deviance follows a chi-square distribution under the null hypothesis that all the explanatory effects in the model are zero. This provides a statistical test for the goodness-of-fit hypothesis and the corresponding p-value can be computed for the statistic.

The difference between the residual and null deviances indicates how much the explanatory variables help to improve the model's fit. Table 5.4 depicts the reduction of 1641.36 in deviance by entering logTD variable into the model. This is the highest possible reduction made among those variables we are interested in. The larger the reduction in deviance, the better the model fits the training data set. Given the logTD variable into the model, the maximum reduction, 156.41, is induced by the UK variable. To determine whether a particular reduction is statistically significant, a p-value is obtained from an analysis of deviance chi-square test (see p-value measure in the Table 5.3).

Penalised residual deviance: When estimating model parameters using the ML approach, adding any parameter, regardless of the significance, decreases the residual deviance. In practice, a model with fewer explanatory variables and interaction terms is usually preferred, given that it has comparable residual deviance to a more complex model. This is because simpler models yield more intuitive justifications and are less likely to over-fit the training dataset. The problem of using residual deviance in model selection can be resolved by introducing a penalty term for the number of parameters in the model.

Two common criteria which are defined as the penalised version of residual deviance are: Akaike Information Criterion (AIC) and Bayes Information Criterion (BIC). Suppose the model contains k explanatory effects. For the j-th observation, let p_j denote the estimated probability of the observed response. A commonly-used quality metric AIC augments the residual deviance measure with the number of explanatory variables and assigns a lower (better) score to simpler models. The AIC is computed by:

$$\text{AIC} = -2\log L + 2(k+1), \tag{5.3}$$

Schwarz (1978) introduced another penalised residual deviance in which the penalty is larger than in the AIC. It is usually called Bayesian Information Criterion (BIC), or in some literatures referred as to Schwarz Criterion. The formula is given as:

$$\text{BIC} = -2\log L + (k+1) \times \log(N), \tag{5.4}$$

where N is the number of observation. The AIC and BIC statistics give two different ways of adjusting the $-2\log$-Likelihood statistic for the number of terms in the model and the number of observations used. These statistics should be used when comparing different models for the same data; lower values of the statistic indicate a more desirable model.

Table 5.7: Model fit statistics for Intercept model and the selected stepwise model fitted

Criterion	Intercept Only	Intercept and Covariates
RDev	8071.09	5816.33
AIC	8073.09	5836.82
BSC	8080.35	5908.92
R^2 Nagelkerke	0	0.32

Generalized Coefficient of Determination: Cox and Snell (1989) proposed the generalization of the coefficient of determination to a more general linear model:

$$R^2 = 1 - \left\{ \frac{L(0)}{L(\widehat{\boldsymbol{\beta}})} \right\}^{\frac{2}{N}} , \tag{5.5}$$

where $L(0)$ is the likelihood of the model with only an intercept parameter, or the null deviance, $L(\boldsymbol{\beta})$ is the likelihood of the specified model, and N is the sample size. The quantity R^2 achieves a maximum of less than one for discrete models, where the maximum is given by

$$R^2_{max} = 1 - \{L(0)\}^{\frac{2}{N}} . \tag{5.6}$$

Nagelkerke (1991) proposes the following adjusted coefficient, which can achieve a maximum value of one:

$$\widetilde{R}^2 = \frac{R^2}{R^2_{max}} . \tag{5.7}$$

Table 5.7 contains the AIC, the BIC and the RDev measures for the difference between the intercept-only model and the fitted model. These model fit statistics can be used to compare different models. The model with smaller values is preferred. The statistics confirm that the general session attributes do not give us sufficient information to explain all purchase behaviour; they explain around 32% of the variation.

5.8 Classification by logistic regression

Logistic regression can be applied as a classification technique to assign each item into the class which is most probable given the realizations of the explanatory variables. For binary response data, the response is either an event or a non-event. From the fitted model, a

Table 5.8: Association of predicted probabilities and observed responses for the stepwise logit model fitted (using test data set). The left hand side measures compute for the cut-point which produces the highest largest positive and true negative simultaneously

True Positive	0.69	Somers's D	0.675
True Negative	0.69	Gamma	0.694
False Positive	0.31	Tau-a	0.151
False Negative	0.31	C-Index	0.835

predicted event probability can be computed for each observation. If the predicted event probability exceeds some cut-point value $z \in [0\,,\,1]$, the observation is predicted to be an event observation; otherwise, it is predicted as a non-event. A 2×2 frequency table can be obtained by cross-classifying the observed and predicted responses. Each cut-point generates a classification table.

5.8.1 Classification assessment

A good model must accurately classify items of the training data set, as well as the test data set. The logit model assigns each item the probability of being 1. Setting a threshold, say 0.5, we classify the items for which the model-calculated probability ≤ 0.5 as 0, and for > 0.5 as 1. There is no definite answer to find the optimal threshold. It depends on the relative costs of obtaining false positives (e.g., mistakenly predicting that a person will purchase when he/she actually does not) versus false negatives (e.g., mistakenly predicting that a person will not purchase online when he/she actually does). If the cost is the same, then an optimal value for the cut-point would be the value which produces the highest *true positive*, the number of correctly predicted events, and *true negative*, the number of correctly predicted non-events, simultaneously.

Table 5.8 contains four measures of association of predicted probabilities and observed responses which helps to assess the predictive ability of a model. We use the test data set to compute these measures. They are based on the number of pairs of observations with different response values, the number of concordant pairs, and the number of discordant pairs, which are also displayed.

A common way of showing the trade-offs of different thresholds is by using an ROC curve, a plot of the true positive rate (the number of true positives over total number of positives) versus the false positive rate (the number of false positives over total number of negatives)

for all possible choices of thresholds. For example, a high threshold will yield a low false positive rate, as only samples with very high probabilities will be classified as positive. We will explain in the next section how the optimum cut-point can be found using the intersection of sensitivity and specificity curves. For the fitted logistic model the optimum cut-point probability is 0.35.

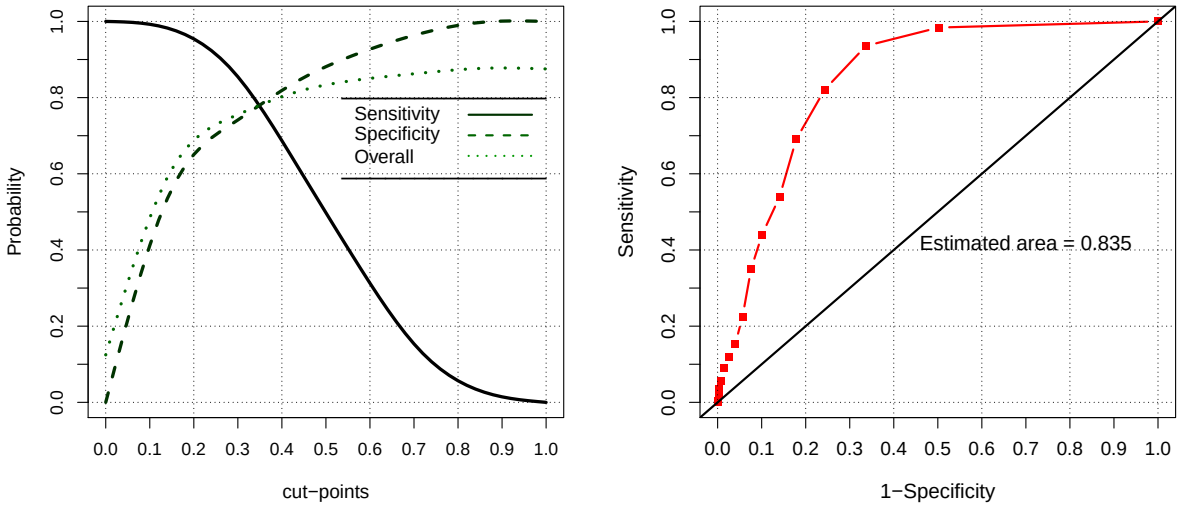
5.8.2 Receiver Operating Characteristic (ROC) Curves

The accuracy of the classification is measured by its *sensitivity*, the ability to predict an event correctly, and *specificity*, the ability to predict a non-event correctly. Sensitivity is computed through the proportion of event responses that were predicted to be events. Similarly, specificity is the proportion of non-event responses that were predicted to be non-events. It is also common to compute three other conditional probabilities: false positive rate, false negative rate, and rate of correct classification. The false positive rate is the proportion of predicted event responses that were observed as non-events. The false negative rate is the proportion of predicted non-event responses that were observed as events. Given prior probabilities, these conditional probabilities can be computed as posterior probabilities using Bayes' theorem.

In the SLC data set, suppose n_1 individuals are observed to make an online purchase. Let this group be denoted by C_1, and let the group of the remaining $n_0 = n - n_1$ individuals who do not make online purchase be denoted by C_0. Significant effects are identified for the sample and a logistic regression model is fitted to the data. For the i-th individual, an estimated probability $\widehat{p}_i$ of the purchasing is calculated. Higher values of the estimated probability are assumed to be associated with the event. A receiver operating characteristic (ROC) curve is constructed by varying the cut-point that determines which estimated event probabilities are considered to predict the event. For each cut-point z, the following measures can be computed to a data set:

$$\begin{aligned}
\mathrm{TP}(z) &= \sum_{i \in C_1} I(\widehat{p}_i \geq z) \\
\mathrm{TN}(z) &= \sum_{i \in C_0} I(\widehat{p}_i < z) \\
\mathrm{FP}(z) &= \sum_{i \in C_0} I(\widehat{p}_i \geq z) \\
\mathrm{FN}(z) &= \sum_{i \in C_1} I(\widehat{p}_i < z),
\end{aligned}$$

Figure 5.6: *The plot for the sensitivity and specificity of the model for different thresholds (left); ROC Curve to show the ability of the model to predict the event (right). The black line represents the model that performs no better than random classification. Results are based on test data set*



where $I(.)$ is the indicator function. TP (True Positive) is the number correctly predicted as buyers, TN (True Negative) is the number correctly predicted as non-buyers, FP (False Positive) is the number falsely predicted as buyers, FN (False Negative) is the number falsely predicted as non-buyers. We consider the proportion of the true positive and true negative events as measures of sensitivity and specificity.

$$\begin{aligned} \text{SENS}(z) &= \frac{\text{TP}(z)}{n_1} \\ \text{SPEC}(z) &= \frac{\text{TN}(z)}{n_0} \\ \text{MSPEC}(z) &= \frac{\text{FP}(z)}{n_0}. \end{aligned}$$

SENS (Sensitivity) function is the proportion of number of true positive events over number of events, and SPEC (Specifity) is computed through fraction of true negative over number of non-events. The value of $1 - \text{SPEC}(z)$ which is the proportion of the number of false positives over the number of non-events.

A plot of the ROC curve is constructed by plotting sensitivity against 1−specificity.

A model with good classification accuracy should have significantly more true positives than false positives at all thresholds. Figure 5.6 (right) shows the predictive ability of the model in comparison with the random classification above. The black line, corresponds to a model of random classification. An area of 0.5 corresponds to a model that performs no better than random classification and an area of 1 is ideal (any area above 0.9 is extremely impressive). The area under the ROC curve quantifies model classification accuracy; the higher the area, the greater the disparity between true and false positives, and the stronger the model in classifying members of the training data set, 0.835 in our study. The area under the ROC curve equals the statistic C-Index. These graphs and corresponding statistics are computed based on test data set.

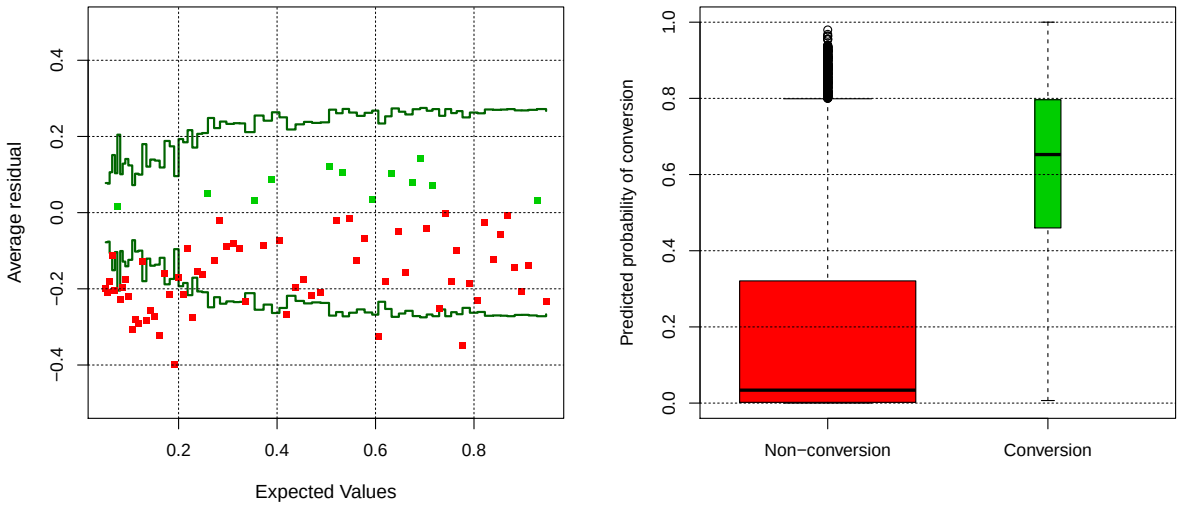
To produce this graph, the parameters of the model are estimated by the training data and then the model is applied to test data to predict the probability of a conversion. We consider those buyers for whom the predicted probability exceeds the specific cut-points. In this case, sensitivity is computed by the ratio of the people who are correctly classified as buyers. Specificity is defined as the ratio of the correctly classified non-buyers. The C-Index is the area under the ROC curve, which can be easily computed as the ratio of correct classification (both buyers and non-buyers).

Figure 5.6 shows the ability of the model to distinguish visitors who purchase (sensitivity), do not purchase (specificity), and a measure of overall ability of true classification (C-Index) using test data set. Choosing the best cut-point for the probability of purchase gives us 78% precision of classification. We may choose a cut-point of 0.25 by which we classify nearly 90% of buyers and 70% of non-buyers correctly.

5.8.3 Model Diagnostics

In this section we use model diagnostic procedures and tests to investigate whether the assumptions of logistic regression fitted on SLC data are satisfied. The residuals for logistic regression are computed as with linear regression - observed response minus expected values. However, as the response is a probability ($0 \leq p_i \leq 1$), depending on whether the response is 0 or 1 residuals are negative or positive respectively. Hence, ordinary residual plots are not useful for logistic regression. The binned residual plot is a diagnostic tool for which the range of the predicted values is divided into several bins, and within each bin the average of the predicted values and average of residuals are computed. Finally, it plots the averages of binned residuals versus average of binned predicted values. Based on assumptions of logistic regression the residuals are independent with mean zero, thus their binned averages also have these properties. As a result, the binned plot is expected

Figure 5.7: *The binned residual plot to check the assumption of independent residuals (left); Boxplot of the predicted probabilities for each response category, Conversion versus Non-conversion visits (right).*



to show a random pattern around the horizontal line at zero if the assumption of the model holds. The binned residual plot also displays more information about the reference distribution by showing error bounds for residuals. These bounds can be computed either using simulation, or if the number of points averaged within each bin is reasonably large, the mean residuals are approximated by the Normal distribution. We can then display the 95% error bounds for each bin (see Gelman and Hill (2006), page 97).

Figure 5.7 (left) displays the residual binned plot for the fitted logistic regression on conversion. We used the function `binnedplot` in the package `arm` to produce the binned plot. It is observed that for low probabilities of conversion, residuals show a systematic discrepancy from zero toward negative values, indicating that the assumption of independent errors is not met. The stepwise lines indicate 95% bounds for residuals. Many binned residuals fall outside the bound for low probabilities and show a model misfit. So, despite the predictive ability of the logistic regression model, there are some issues regarding the assumptions of the model. Figure 5.7 (right) shows the boxplot of the predicted probabilities based on the logit model for the two groups of conversion and non-conversion visits. The width of boxes is proportional to the number of observations/visits in each group. It can be seen that the distribution of probabilities in the non-conversion group is heavily skewed toward zero.

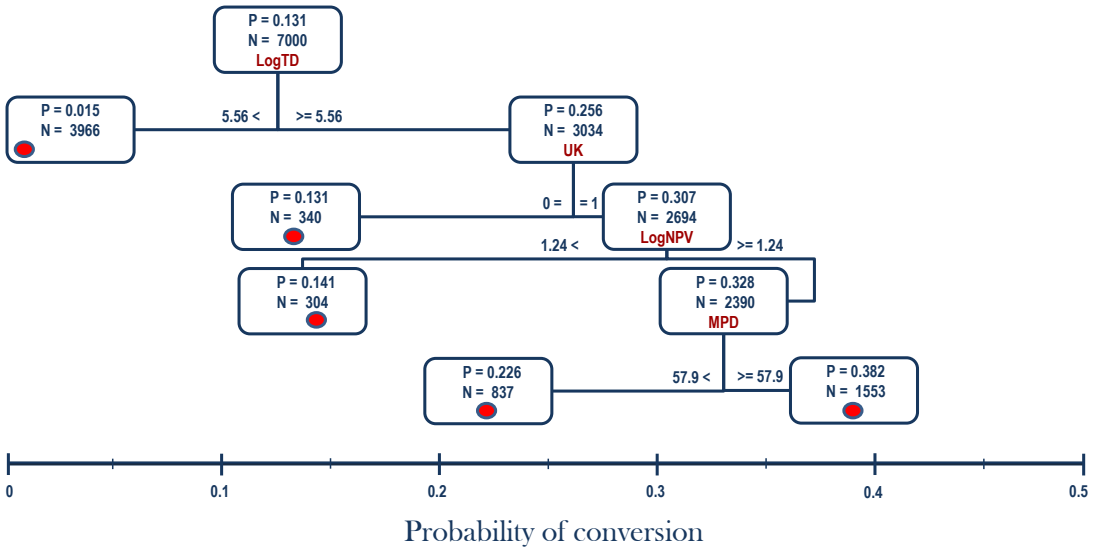
5.9 Classification and Regression Tree

There are additional alternative approaches to logistic regression that can be used to classify the visits in terms of conversion. Although we do not aim to investigate different classification approaches and choose the best one, an alternative approach helps to benchmark the predictive ability of logistic regression. In this section we use the classification and regression trees (CART) method, as an alternative nonparametric data mining algorithm to the standard logistic regression model (Austin, 2007). The CART algorithm is a widespread approach which is well-known for its ease of interpretation. The result of the CART model is usually represented by a decision tree. Decision trees are composed of a set of rules based on variables in the modelling data set so that rules are selected to achieve the best split of response variable. A decision tree splits a *node* or the explanatory variable so that it best differentiates between the categories of the target variable. The same process is applied to each *child node*. Splitting stops when no further significant splits are found, or some pre-set stopping rules are met. As an alternative, algorithm data is split as much as possible and then non-significant branches are ignored, known as *pruning*. Each branch of the tree ends in a *terminal node*, where there is no more split. Each terminal node is uniquely defined by a set of rules. Each observation falls into exactly one terminal node. For detailed technical discourses of the CART methodology, also referred to as binary recursive partitioning, see Breiman et al. (1984).

We use the same training data set from the SLC data as used to estimate the parameters of logistic regression. This enables us to compare the fitted logistic regression and the CART outputs in terms of predictive ability, as well as significancy of the predictors. The results of CART is displayed in a tree structure in Figure 5.8. The x-axis defines the probability of incidence (conversion in our study). So the location of each node with respect to the x-axis shows the probability of conversion in the subgroups identified by the node. The most significant association with conversion is found for the logTD variable with a cut-off at 5.56 (equivalent to 4.5 minutes). It can be seen that the chance of conversion is very low for visiting sessions lasting less than 4.5 minutes, 0.015. The odds ratio of conversion for the long sessions (more than 4.5 minutes) versus short session visits is 25 to 1.

In the second step, the procedure is repeated within the subgroups identified in step one, for sessions which last more than 4.5 minutes. The most significant split is obtained by UK versus non-UK visitors. The odds ratio for the subgroup of the long visit sessions on conversion is 4 to 1 in favour of UK visits. The last split is found at level four by splitting the average time spent on pages, MPD, for 2390 visits. This subgroup is split into 837 visits with an average page time duration of below 57 seconds with conversion probability

Figure 5.8: *Classification and regression tree (CART) analysis of general clickstream data on conversion.*

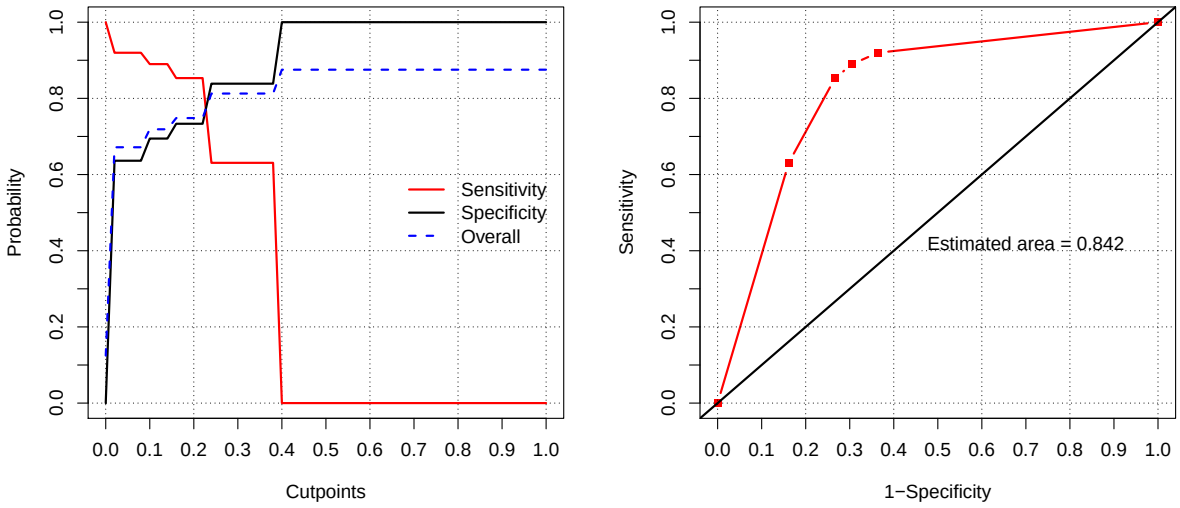


of 0.236, and a second group of 1553 with an average page time duration of above 57 seconds with a conversion probability of 0.382. The tree building procedure stops here as no further significant splits can be found or further splitting would create subgroups of insufficient size. For more information about stopping rules and pruning see Breiman et al. (1984).

We use the test data to produce the ROC curve of the CART, see Figure 5.9 (left), to assess the predictive ability of the model in comparison with the random classification above. The optimal cut-off is located at 0.22, where both the sensitivity and specificity are the same and equal to 0.78. This value is very close to equivalent optimum sensitivity and specificity of logistic regression. Figure 5.9 (right) shows the area under the ROC curve for the CART model as a metric for model classification accuracy. This value is also very similar to the accuracy achieved by logistic regression. This way, neither the logistic model nor the CART out-performs each other in terms of predictive ability.

When the CART results are compared to the findings of logistic model, in both cases the strongest relationships are identified by logTD and UK. Unlike the logistic regression, the Hour.C2 factor has not been chosen at this level of pruning for the CART model. However, CART is also found in logNPV and MPD to be a relevant predictor for conversion which

Figure 5.9: *The plot for the sensitivity and specificity of the CART model for different thresholds (left); ROC Curve to show the ability of the model to predict the event (right). The black line represents the model that performs no better than random classification.*



has not been entered in the final logistic regression by the model selection step. This can be justified by the multi-collinearity between the predictors. For example, the correlation between the logTD and MPD variables is 0.482. It is also known that logistic regression models generally focus more on the relative statistical significance, whilst CART results tend to find the absolute effects (Muller and Möckel, 2008).

However, repeating the CART for different subsets of SLC data showed that only the first and second roots of the CART (logTD and UK/Non-UK) are stable, whilst the selected logistic is robust and did not change. This is in addition to the fact that different definitions of cut-offs, split and stopping criteria, as well as pruning procedures may present different classification trees.

5.10 Summary and discussion

In this chapter we used the logit model to describe the association between general clickstream information concerning visits and whether a visitor will engage in online-purchasing behaviour during his visit to the website. This model provides a predictive tool for an e-commerce web owner that helps to infer the visitor behaviour, and as a

result, to improve the online conversion rates of the website.

Past research has already incorporated the presented clickstream information to examine the relationship with purchase propensity. However, they all considered a main effect model and in some cases quadratic forms of variables in their studies. This study not only incorporates the proposed explanatory variables into the model but also allows the interaction terms to be added to the model if they contribute for better goodness-of-fit.

The results show that the performance of the model increased significantly by considering the interaction terms into the model. Unfortunately, the SLC data does not contain the detailed clickstream variables and demographics to allow us to compare the performance of the model with previous studies. All three variable selection procedures (forward selection, backward elimination, and stepwise) that were applied suggest the same model. We consider the selected subset as the most important variable to describe online purchasing behaviour.

The most important variables that result from the selection techniques are the logarithm of the session time duration, whether the visitor is from the UK, the number of former visits to the website, the logarithm of the former time spent on the website, The interaction of session time duration and time previously spent on the website, whether or not visitors arrive at the website from Google, and finally the interaction of domestic visit from the UK and linking to the website from Google.

There are some explanatory variables that are significant when performing a single logistic regression Table 5.3, which were not entered into the final model (e.g., the number of pages visited). This is mostly because of the presence of multi-collinearity. The correlation matrix of the explanatory variables in the subset of chosen variables shows that there is not a large multi-collinearity which would be present when incorporating all of the explanatory variables.

The model presented in this chapter offers a more in-depth investigation of conversion behaviour based on general visit information compared to models without the interaction terms. This results in a higher predictive ability and a better way of classifying customers concerning their purchase behaviour on the Internet. This is a significant contribution toward understanding the features that control a visitor's decision to make a purchase or not. Moreover, we can limit the number of necessary inputs based on the different selection techniques.

The logit model can help to approximate the probability of whether or not a purchase is made during the visit using the set of predictors used in the model. The ease of inter-

pretation of logit is an important advantage over other methods such as neural networks. Once the coefficient parameters are estimated, this model allows us to obtain a conditional probability estimate of purchase. The probability approximation can be used to rank customers in terms of their probability of purchase.

From the digital marketing point of view, using interaction and squared terms in the logistic regression for a small number of generic web browsing features enabled us to reach reasonable classification power. The logistic regression model provides a mathematical equation that can easily estimate the probability of online purchase for each visitor while he/she is browsing the website. This way, the web owner would be able to identify high potential visitors, in terms of conversion tendency, and generate leads for suitable targeting actions. In a web-focused marketing solution, the targeting action might help to keep the customer in the website before leaving to find another competitor.

There are additional alternative approaches including neural networks, categorical principal component analysis (CATPCA) and further developments of CART models such as multiple additive regression trees (MART) and multivariate adaptive regression (MARS) which can be used to find the *best* approach for predicting conversion visits. Choosing the best approach discussed in the literature would go beyond the scope of this thesis, and is not our aim. The CART approach was chosen here mainly because of its ease of interpretation and its relatively widespread use. The sensitivity and specificity values show that both logistic regression and CART are suitable for classification purposes. So the CART model is an alternative for classification when the required assumption for the parametric model such as logistic regression is not met. However, CART is not very stable and is subject to change by using different sets of data.

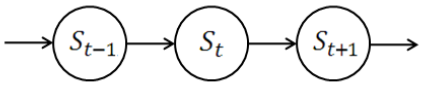
Chapter 6

Bayesian Mixture of Hidden Markov Models

6.1 Introduction

This chapter essentially provides the theoretical background for the Bayesian analysis of Mixture of Hidden Markov models (MixHMM). The application, modelling internet browsing behaviour in an e-commerce website, will be discussed in the next chapter. As the aim is to model web-page traversal we focus on the output of hidden Markov Models with discrete nominal distributions. We assume that the number of components and the number of regimes for the hidden Markov models are known. The implementation of the MixHMM will be extended to the Bayesian context through direct Gibbs sampling and forward-backward Gibbs sampling methods. For this reason we review the relevant methods in mixture models and Hidden Markov models such as forward-backward recursion. Note that the classical methods of HMM are used in the Bayesian context to improve the mixing in the Gibbs sampling from the conditional distribution. We also examine the label-switching problem in the application of the MCMC method for MixHMM which is an issue for both cases of mixture models and HMMs. Using Gibbs sampling for the MixHMM, as a high dimensional model, one might face a high level of autocorrelation for samples. use thinning to provide an independent sample from the marginal posterior distribution of the model parameters. The performance of the model is assessed over an artificial navigation pattern. The contribution in this chapter is to implement the MixHMM in the Bayesian framework. see section 6.8 for a simulation study and example to illustrate the following theory. See chapter 7 to see an application of the theory to our actual data set.

Figure 6.1: *Graphical representation of a Markov model*



6.2 Markov Model

A Markov model (Markov chain) is a stochastic process with the Markov property. The term *stochastic* means that all transitions between different states of the process are probabilistic, and having the Markov property means that, given the present state, future states are independent of the past. Mathematically, The Markov chain is defined as a sequence of random variables $\{S_t;\ t = 1, 2, \ldots, T\}$ for which $S_t \in \mathcal{S} = \{1, 2, \ldots, S\}$. The Markov property of a Markov chain can be expressed by:

$$\begin{aligned} \Pr\{S_t \mid S_{t-1}, \ldots, S_1\} &= \Pr\{S_t | S_{t-1}\} \\ &= \Pr\{S_t = s' | S_{t-1} = s\} \\ &= \gamma_t(s, s'). \end{aligned} \tag{6.1}$$

The conditional independence can be denoted by $S_t \perp\!\!\!\perp S_1, \ldots, S_{t-2} \,|\, S_{t-1}$. A graphical representation of a Markov model has been depicted in Figure 6.1 using a directed acyclic graph (DAG). A DAG consists of an arrangement of connected nodes and edges, where nodes represent the unknown or observed quantities. The edges represent the dependencies between nodes. The conditional distribution of each node, given the value of all other nodes, depends only on the nodes to which it is connected.

If $\gamma_t(s, s') = \gamma(s, s')$ for every t where $s, s' = 1, 2, \ldots, S$ then $\{S_t;\ t = 1, 2, \ldots, T\}$ is called a homogeneous Markov chain. The probability of transition between states is usually represented in a matrix form, $\Gamma = [\ \gamma(s, s')\ ]_{S \times S}$, known as the *transition matrix*. The probability associated to the element $(i\ ,\ j)$, the i-th row and j-th column, shows the probability of transition from state i to the state j. Let $\pi_t(s) = \Pr\{S_t = s\}$ denote the probability distribution over the S_t. The probability of being at state s at time $t = 1$, $\pi_1(s)$, is called the *initial state probability*. One may use vector notation to represent the state probability distribution:

$$\boldsymbol{\pi}_t = [\pi_t(1), \pi_t(2), \ldots, \pi_t(S)]. \tag{6.2}$$

The chain is called *stationary* if for every t, $\boldsymbol{\pi}_t = \boldsymbol{\pi}$. That is, the distribution of the hidden state regardless of the time remains the same.

6.3 Hidden Markov Model

A hidden Markov Model (HMM) is a Markov chain in which the states are unobserved, and the inference about the states is essentially based on visible observed outputs which depends on the hidden states. There exist many phenomena and systems based on HMM, and consequently they have been used in a variety of fields such as signal processing (Rabiner, 1989; Juang and Rabiner, 1991), finance and econometrics (Albert and Chib, 1993), genetics (Churchill, 1989), and many other disciplines (Castellano and Scaccia, 2007). HMM is also related to other classes of stochastic models. It can be considered as a mixture model in which the mixing distribution is a finite state Markov chain (Scott, 2002). As a special case, when all rows of the transition matrix are equal, HMM converts into a finite mixture model (Everitt and Hand, 1981). A common example of a hierarchical HMM with observable outputs from the Normal distribution is the conditionally Gaussian linear state-space model (West and Harrison, 1997).

Throughout this section we mostly use Rabiner (1989), a well-known introduction for HMM, to illustrate the basic concepts of the HMM. Specifically, the likelihood for the model is given and how the resulting likelihood function can be optimized with the EM-algorithm is shown. This involves maximising the likelihood function with respect to the parameters of the model. In this section the computation is given for some known family of probability distributions.

Note that in a regular Markov model the states are directly visible to the observer. Therefore, the state transition probabilities are the only parameters of the model, but for a HMM in addition to the transition probabilities, the probabilities of observing an output given the hidden state are unknown parameters of the model. Each state variable has a probability distribution over the possible outputs.

Mathematically, the hidden Markov model is a sequence of hidden variables $\{S_t;\ t = 1, 2, \ldots, T\}$, usually known as state variables, with corresponding observable random variables $\{Y_t; t = 1, 2, \ldots, T\}$, usually referred to as output variables. Both states and outputs are indexed by time t. The hidden Markov chain $\{S_t\}$ can take values from the set of all possible states $\mathcal{S} = \{1, 2, \ldots, S\}$ at any time. The realization s_t of the variable S_t is called the state (or regime in some applications) of the chain at time t, of course taking values from $\mathcal{S}$. The hidden Markov model satisfies the Markov property over the hidden state variable: given the value of hidden variable S_{t-1}, the conditional probability distribution of the hidden variable S_t at time t depends only on the value of S_{t-1}. Additionally, The output variable Y_t only depends on the hidden variable at time t, S_t. When there is only one state available, $|\mathcal{S}| = 1$, the HMM is reduced to a simple random walk.

Figure 6.2: *Graphical representation of a HMM. The conditional distribution of each node, given the value of all the other nodes depends only on the nodes to which it is connected by an edge.*

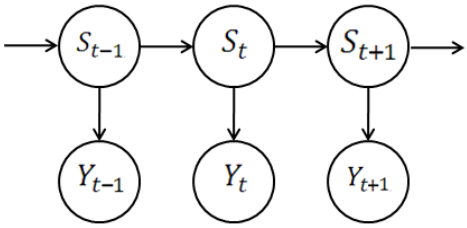


Figure 6.2 illustrates the dependency structure in a HMM by a DAG. For ease of notation, hereafter we denote the history of the process until time t by $Y_{(t)}$, and equivalently for hidden states $S_{(t)}$. This can be seen from the DAG represented in Figure 6.2 that $Y_t \perp\!\!\!\perp S_{(t-1)}, Y_{(t-1)} \,|\, S_t$, that is:

$$\Pr\{Y_t|Y_{(t-1)}, S_{(t)}, \boldsymbol{\theta}\} = \Pr\{Y_t|S_t, \boldsymbol{\theta}\}. \tag{6.3}$$

Throughout this chapter, the conditional probability distribution function for a discrete random variable Y_t will be denoted by:

$$p_s(y_t) = \Pr\{Y_t = y_t|S_t = s, \boldsymbol{\theta}\}. \tag{6.4}$$

When Y_t is a continuous random variable, the probability function is replaced with the density function, $f_s(y_t)$. Where Λ is the set of parameters relevant to the observed probability distribution $p_s(y_t)$, then a hidden Markov model is usually denoted by the triplet $\Theta = (\boldsymbol{\pi}, \Gamma, \Lambda)$ collectively. The elements of the matrix Γ are referred to as transition probabilities, and the elements of the Λ are known as *emission probabilities*. Throughout this chapter, instead of Θ notation we use a vector of all parameters of the model denoted by $\boldsymbol{\theta}$.

6.3.1 Likelihood in HMM

For a given sequence of the observations, the ML estimate of the parameters, $\Theta = (\boldsymbol{\pi}, \Gamma, \Lambda)$, can be derived by maximising the likelihood function of HMM. Consider the sequence of observations $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_T)$ and the sequence of hidden states $\boldsymbol{S} = (S_1, S_2, \ldots, S_T)$, and their corresponding realizations by $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ and $\boldsymbol{s} =$

$(s_1, s_2, \ldots, s_T)$ respectively. The likelihood function for a HMM is given by

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &\propto \Pr\{\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{\theta}\} && (6.5)\\
&= \sum_{\mathcal{S}^T} \Pr\{\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{\theta}\} && (6.6)\\
&= \sum_{\mathcal{S}^T} \Pr\{\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{S} = \boldsymbol{s}, \boldsymbol{\theta}\} \Pr\{\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{\theta}\} && (6.7)\\
&= \sum_{\mathcal{S}^T} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{S}) && (6.8)
\end{aligned}$$

where $\mathcal{S}$ is the set of all possible hidden states, and $\mathcal{S}^T$ denotes the set of all possible combinations of hidden states for the sequence of observations of length T. In view of the fact that HMM observations are conditional on the hidden states of the chain, the maximisation needs to be implemented over the complete likelihood function, $L\{\boldsymbol{\theta}, \boldsymbol{S}\}$, in which the hidden states are considered to be known. Considering the Markov property in HMM we have

$$\begin{aligned}
\Pr(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{\theta}) &= \Pr(S_1 = s_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \Pr(S_t = s_t | S_{t-1} = s_{t-1}, \boldsymbol{\theta})\\
&= \pi_1(s_1)\gamma(s_1, s_2)\gamma(s_2, s_3) \cdots \gamma(s_{T-1}, s_T), && (6.9)
\end{aligned}$$

and the distribution of the observed variable Y_t only depends on the hidden state S_t, that is

$$\Pr\{\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{S} = \boldsymbol{s}, \boldsymbol{\theta}\} = \prod_{t=1}^{T} \Pr\{Y_t = y_t \mid S_t = s_t, \boldsymbol{\theta}\} = \prod_{t=1}^{T} p_{s_t}(y_t). \tag{6.10}$$

The complete likelihood, $L_c(\boldsymbol{\theta}) = L(\boldsymbol{\theta}, \boldsymbol{S})$, in HMM for a given sequence of $\boldsymbol{S} = \boldsymbol{s}$ is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{S} = \boldsymbol{s}) = \pi_1(s_1)\gamma(s_1, s_2)\gamma(s_2, s_3) \cdots \gamma(s_{T-1}, s_T) \prod_{t=1}^{T} p_{s_t}(y_t). \tag{6.11}$$

Maximisation of the log-likelihood

Maximising the complete likelihood when the length of the sequence of observations is large may cause an overflow problem. In practice, the logarithm of the likelihood is a suitable alternative for computational purposes. Having defined the following indicator functions:

$$\begin{aligned}
u_t(s) &= \begin{cases} 1 & \text{if } S_t = s \\ 0 & \text{if } S_t \neq s \end{cases}\\
v_t(s, s') &= \begin{cases} 1 & \text{if } S_{t-1} = s \text{ and } S_t = s' \\ 0 & \text{if } S_{t-1} \neq s \text{ or } \quad S_t \neq s' \end{cases}
\end{aligned} \tag{6.12}$$

The logarithm of the complete likelihood (6.11) known as the log-likelihood function $\ell_c(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}, \boldsymbol{S})$ can be rewritten as a sum of three terms,

$$\ell(\boldsymbol{\theta}, \boldsymbol{S}) = \sum_{s=1}^{S} u_1(s) \ln \pi_1(s) + \sum_{s=1}^{S}\sum_{s'=1}^{S} \left(\sum_{t=2}^{T} v_t(s, s') \right) \ln \gamma(s, s') + \sum_{s=1}^{S}\sum_{t=1}^{T} u_t(s) \ln p_s(y_t). \tag{6.13}$$

which can be maximised separately to find the maximum of the (6.13). We illustrate how to find the maximum likelihood estimate of the transition and emission parameters for a HMM by maximisation of the complete likelihood function in the following theorems. These results have been already proved, see Harte (2010) page 6, but we explain the proofs in more details to developed the results for the MixHMM in §6.4.

Theorem 6.1 *The first term of the complete log-likelihood function in* (6.13) *is maximised by the following initial state probabilities:*

$$\widehat{\pi}_1(s) = \frac{u_1(s)}{\sum_{s=1}^{S} u_1(s)}.$$

Proof In order to maximise the first term of the complete likelihood (6.13), one needs to consider the constraint of $\sum_{s=1}^{S} \pi_1(s) = 1$. Defining the function F,

$$F = \sum_{s=1}^{S} u_1(s) \ln \pi_1(s) + \eta \left(1 - \sum_{s=1}^{S} \pi_1(s) \right), \tag{6.14}$$

where η is the Lagrange multiplier. The derivatives of F with respect to $\pi_1(s)$ for $s = 1, 2, \ldots, S$,

$$\frac{\partial F}{\partial \pi_1(s)} = \frac{u_1(s)}{\pi_1(s)} - \eta, \tag{6.15}$$

implies that $\eta = \sum_{s=1}^{S} u_1(s)$. Therefore,

$$\widehat{\pi}_1(s) = \frac{u_1(s)}{\sum_{s=1}^{S} u_1(s)}. \tag{6.16}$$

□

Theorem 6.2 *The second term of the complete log-likelihood function in* (6.13) *is maximised by the following transition probabilities:*

$$\widehat{\gamma}(s, s') = \frac{\sum_{t=2}^{T} v_t(s, s')}{\sum_{t=2}^{T} u_t(s)}.$$

Proof For the second term of equation (6.13), we need to consider that there are S constraints, $\sum_{s'=1}^{S}\gamma(s,s')=1$, corresponding to the rows of the transition matrix. So, we define the function G, .

$$G=\sum_{s=1}^{S}\sum_{s'=1}^{S}\left(\sum_{t=2}^{T}v_t(s,s')\right)\ln\gamma(s,s')+\sum_{s=1}^{S}\eta_s\left(1-\sum_{s'=1}^{S}\gamma(s,s')\right), \tag{6.17}$$

where $\eta_1,\eta_2,\ldots,\eta_S$ are the Lagrange multipliers, then

$$\frac{\partial G}{\partial\gamma(s,s')}=-\eta_s+\frac{1}{\gamma(s,s')}\sum_{t=2}^{T}v_t(s,s'). \tag{6.18}$$

Hence, if we consider $-\eta_s\gamma(s,s')+\sum_{t=2}^{T}v_t(s,s')=0$ then

$$\sum_{s'=1}^{S}\Big(-\eta_s\gamma(s,s')+\sum_{t=2}^{T}v_t(s,s')\Big)=0. \tag{6.19}$$

Since $\sum_{s'=1}^{S}\gamma_1(s,s')=1$ then $\eta_s=\sum_{s'=1}^{S}\sum_{t=2}^{T}v_t(s,s')$ so that

$$\widehat{\gamma}(s,s') = \frac{\sum_{t=2}^{T}v_t(s,s')}{\sum_{s'=1}^{S}\sum_{t=2}^{T}v_t(s,s')} \tag{6.20}$$

$$= \frac{\sum_{t=2}^{T}v_t(s,s')}{\sum_{t=2}^{T}u_t(s)}. \tag{6.21}$$

□

We also need to maximise the last term of the complete likelihood:

$$\sum_{s=1}^{S}\sum_{t=1}^{T}u_t(s)\ln p_s(y_t). \tag{6.22}$$

This part of the complete log-likelihood depends on the probability distribution of the observed process. Harte (2010) illustrate this step for a few of the most commonly-used distributions in HMM. We explain the computations in more depth for discrete nominal, Poisson, Gamma, Binomial, and Normal distributions and result will be developed for the MixHMM.

Theorem 6.3 (Discrete nominal distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the parameter of the discrete nominal distribution:*

$$\widehat{p}_s(y)=\frac{\sum_{t=1\wedge Y_t=y}^{T}u_t(s)}{\sum_{t=1}^{T}u_t(s)},$$

where $p_s(y)$ is the probability of observing y for the group of observation in the hidden state s.

Proof In this case we first define

$$w_t(y,s) = \begin{cases} 1 & \text{if } Y_t = y \text{ and } S_t = s \\ 0 & \text{if } Y_t \neq y \text{ or } \quad S_t \neq s \end{cases} \tag{6.23}$$

where $y \in \mathcal{S}_y$ is the set of all possible values of Y_t. This way, for a homogeneous HMM the equation (6.22) can be rewritten as:

$$\sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{y\in\mathcal{S}_y} w_t(y,s)\ln p_s(y). \tag{6.24}$$

Considering the constraints over the parameter space, the function H needs to be maximised.

$$H = \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{y\in\mathcal{S}_y} w_t(y,s)\ln p_s(y) + \sum_{s=1}^{S}\eta_s\Big(1 - \sum_{y\in\mathcal{S}_y} p_s(y)\Big). \tag{6.25}$$

The derivative of H with respect to the parameters of the probability distribution over Y_t, the probability of the observation in this case, provides the ML estimate of the parameters.

$$\frac{\partial H}{\partial p_s(y)} = \sum_{t=1}^{T}\frac{w_t(y,s)}{p_s(y)} - \eta_s. \tag{6.26}$$

The solution of the equation $-\eta_s p_s(y) + \sum_{t=1}^{T} w_t(y,s) = 0$ gives us the values for the parameter space which maximise the function H. Note that $\sum_{y\in\mathcal{S}_y} p_s(y) = 1$, so summation over $\mathcal{S}_y$ gives us the η_s:

$$\eta_s = \sum_{y\in\mathcal{S}_y}\sum_{t=1}^{T} w_t(y,s) = \sum_{t=1}^{T} u_t(s). \tag{6.27}$$

so that

$$\widehat{p}_s(y) = \frac{\sum_{t=1}^{T} w_t(y,s)}{\sum_{y\in\mathcal{S}_y}\sum_{t=1}^{T} w_t(y,s)} \tag{6.28}$$

$$= \frac{\sum_{t=1\wedge Y_t=y}^{T} u_t(s)}{\sum_{t=1}^{T} u_t(s)}. \tag{6.29}$$

□

Therefore, the ML estimate of $p_s(y)$ is given by the empirical probability of observing y for the group of observation in the hidden state s. These probabilities can be summarized into a matrix notation $\Lambda = [\, p_s(y)\,]_{|\mathcal{S}_y|\times S}$, as the set of all emission probabilities.

Theorem 6.4 (Poisson distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the parameter of the Poisson distribution:*

$$\widehat{\theta}_s = \frac{\sum_{t=1}^{T} u_t(s) y_t}{\sum_{t=1}^{T} u_t(s)},$$

where θ_s is the rate parameter of the Poisson distribution for the hidden state s.

Proof For the Poisson distribution

$$\Pr\{Y_t = y \mid S_t = s, \boldsymbol{\theta}\} = \frac{\theta_s^y}{y!}\exp\{-\theta_s\} \tag{6.30}$$

so the third part of the complete log-likelihood (6.22) is a function of the rate parameter of the Poisson distribution. The partial derivatives give the ML estimate of the parameters. Let

$$H = \sum_{s=1}^{S}\sum_{t=1}^{T} u_t(s)\Big(y_t \ln\theta_s - \ln(y_t!) - \theta_s\Big), \tag{6.31}$$

and so

$$\frac{H}{\partial\theta_s} = \frac{1}{\theta_s}\sum_{t=1}^{T} u_t(s)\Big(y_t - \theta_s\Big), \tag{6.32}$$

hence

$$\widehat{\theta}_s = \frac{\sum_{t=1}^{T} u_t(s) y_t}{\sum_{t=1}^{T} u_t(s)}. \tag{6.33}$$

The set of parameters is the vector of $\Lambda = [\theta_1, \theta_2, \ldots, \theta_S]$. So the ML estimate of the rate parameters is the average of the observed values in the group of observations which are in the hidden state s.

Theorem 6.5 (Exponential distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the parameter of the Exponential distribution:*

$$\widehat{\theta}_s = \frac{\sum_{t=1}^{T} u_t(s)}{\sum_{t=1}^{T} u_t(s) y_t}.$$

where θ_s is the rate parameter of the Exponential distribution for the hidden state s.

Proof In this case

$$f_{Y_t}(y \mid S_t = s, \boldsymbol{\theta}) = \theta_s \exp\{-\theta_s y\}. \tag{6.34}$$

That is, the third part of the complete log-likelihood (6.22) is a function of rate parameter of the Exponential distribution. Let

$$H = \sum_{s=1}^{S}\sum_{t=1}^{T} u_t(s)\Big(\ln\theta_s - \theta_s y_t\Big), \tag{6.35}$$

and the partial derivatives provide the ML estimates of the parameters.

$$\frac{H}{\partial\theta_s} = \sum_{t=1}^{T} u_t(s)\Big(\frac{1}{\theta_s} - y_t\Big), \tag{6.36}$$

hence

$$\widehat{\theta}_s = \frac{\sum_{t=1}^{T} u_t(s)}{\sum_{t=1}^{T} u_t(s) y_t}. \tag{6.37}$$

The set of parameters is the vector of $\Lambda = [\theta_1, \theta_2, \ldots, \theta_S]$. The ML estimate of the parameter for exponential variables appears to be the classical estimate of the inverse of the average value of observations, for each group of observations in the hidden state s. □

Theorem 6.6 (Binomial distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the parameter of probability of success in the Binomial distribution:*

$$\widehat{\theta}_s = \frac{\sum_{t=1}^{T} u_t(s) y_t}{\sum_{t=1}^{T} u_t(s) n_t}.$$

where θ_s is the probability of success in the Binomial distribution for the hidden state s.

Proof In this case

$$\Pr\{Y_t = y_t \mid S_t = s, \boldsymbol{\theta}\} = \binom{n_t}{y_t} \theta_s^{y_t} (1-\theta_s)^{n_t - y_t} \tag{6.38}$$

Let

$$H = \sum_{s=1}^{S} \sum_{t=1}^{T} u_t(s) \left[\ln \binom{n_t}{y_t} + y_t \ln \theta_s + (n_t - y_t) \ln(1-\theta_s) \right]. \tag{6.39}$$

Now the partial derivatives give us the ML estimates of the parameters.

$$\frac{H}{\partial\theta_s} = \sum_{t=1}^{T} u_t(s)\Big(\frac{y_t}{\theta_s} - \frac{n_t - y_t}{1-\theta_s}\Big), \tag{6.40}$$

hence

$$\widehat{\theta}_s = \frac{\sum_{t=1}^{T} u_t(s) y_t}{\sum_{t=1}^{T} u_t(s) n_t}. \tag{6.41}$$

□

The set of parameters is the vector of $\Lambda = [\theta_1, \theta_2, \ldots, \theta_S]$. The ML estimate is given by the total number of successes over the total number of experiments, separately for each group of observations in the hidden state s .

Theorem 6.7 (Normal distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the mean and variance parameters of the Normal distribution:*

$$\begin{aligned}\widehat{\mu}_s &= \frac{\sum_{t=1}^{T} u_t(s) y_t}{\sum_{t=1}^{T} u_t(s)} \\ \widehat{\sigma}_s^2 &= \frac{\sum_{t=1}^{T} u_t(s)(y_t - \widehat{\mu}_s)^2}{\sum_{t=1}^{T} u_t(s)}.\end{aligned}$$

where μ_s and σ_s^2 are respectively the mean and variance parameters of the Normal distribution for the hidden state s.

Proof In this case

$$f_{Y_t}(y_t|S_t = s, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\Big(- \frac{1}{2\sigma_s^2}(y_t - \mu_s)^2\Big). \tag{6.42}$$

Let

$$H = \sum_{s=1}^{S}\sum_{t=1}^{T} u_t(s)\Big(- \frac{1}{2}\ln 2\pi\sigma_s^2 - \frac{1}{2\sigma_s^2}(y_t - \mu_s)^2\Big), \tag{6.43}$$

and the partial derivatives provide the ML estimates of the parameters.

$$\frac{H}{\partial\mu_s} = \sum_{t=1}^{T} u_t(s)\Big(y_t - \mu_s\Big) \tag{6.44}$$

$$\frac{H}{\partial\sigma_s^2} = \sum_{t=1}^{T} u_t(s)\Big(- \frac{1}{2\sigma_s^2} + \frac{1}{2(\sigma_s^2)^2}(y_t - \mu_s)^2\Big). \tag{6.45}$$

Hence

$$\widehat{\mu}_s = \frac{\sum_{t=1}^{T} u_t(s) y_t}{\sum_{t=1}^{T} u_t(s)} \tag{6.46}$$

$$\widehat{\sigma}_s^2 = \frac{\sum_{t=1}^{T} u_t(s)(y_t - \widehat{\mu}_s)^2}{\sum_{t=1}^{T} u_t(s)}. \tag{6.47}$$

The set of parameters is the vector of $\Lambda = \{(\mu_s, \sigma_s^2), s = 1, 2, \ldots, S\}$. So, the ML estimate μ and σ_s are respectively the sample mean and variance of the observations which are in the state of s.

□

Having the completed likelihood function maximised enable us to find the the ML estimates of the parameter in HMM using the EM algorithm which will be reviewed later in §6.3.4. This algorithm uses the value of the complete likelihood function $\ell_c(\boldsymbol{\theta})$ in a recursive formula, and so it needs to be computed as a part of the parameter estimation step.

6.3.2 Likelihood Recursion

In this section, we review the recursive method which provides the value of the likelihood, at the same time as recursive estimation is implemented. However, this method is useful in other statistical procedures such as Metropolis-Hastings algorithm, or model selection, when it is required to find the likelihood value (Scott, 2002). The likelihood function (6.8) is defined by the sum over elements of $\mathcal{S}^T$. In practice, even for a moderate number of states S, evaluating the likelihood function by direct evaluation is computationally expensive. Hence, one needs a method to compute the likelihood in an efficient way. The so-called *Likelihood recursion algorithm* facilitates the calculation of the likelihood by decreasing the number of steps to $O(S^2T)$ steps which is markedly smaller than $O(S^T)$.

The likelihood recursion method is carried out based on *forward probabilities*. The forward probability $\alpha_t(s)$ is referred to as the joint probability of the partial observation sequence until time t and the state event S_t at time t, that is

$$\alpha_t(s) = \Pr\{Y_1 = y_1, Y_2 = y_2, \ldots, Y_t = y_t, S_t = s \mid \boldsymbol{\theta}\}, \tag{6.48}$$

where $t = 1, 2, \ldots, T$ and $s = 1, 2, \ldots, S$. This probability can be calculated using a forward recursive procedure. The forward probabilities in equation (6.48) can be formulated by the following summation form (see Harte, 2010, pg. 3 for detailed proof). This involves initialising the forward probability $\alpha_1(s) = \pi_1(s)\, p_s(y_1)$ and then updating the next forward probabilities through induction steps

$$\alpha_t(s) = \sum_{s'=1}^{S} \alpha_{t-1}(s')\gamma(s', s)p_s(y_t). \tag{6.49}$$

It is more convenient to represent the recursive formula via matrix notation. Having let $\boldsymbol{\alpha}_t = [\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(S)]$ and $D_t = \text{diag}\{p_1(y_t), p_2(y_t), \ldots, p_S(y_t)\}$, then (6.49) can be written as

$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}\, \Gamma\, D_t, \tag{6.50}$$

where the starting value for the recursive formula is $\boldsymbol{\alpha}_1 = \boldsymbol{\pi}_1\, D_1$. similarly

$$\boldsymbol{\alpha}_t = \boldsymbol{\pi}_1 D_1 (\Gamma D_2)(\Gamma D_3) \cdots (\Gamma D_t). \tag{6.51}$$

The computation is iterated over $t = 1, 2, \ldots, T$. Finally, the summation of $\alpha_T(s)$

$$\alpha_T(s) = \Pr\{Y_1 = y_1, Y_2 = y_2, \ldots, Y_T = y_T, S_T = s \mid \boldsymbol{\theta}\} \tag{6.52}$$

over possible values of $\mathcal{S}$ provides the desired likelihood value (Rabiner, 1989).

$$\mathcal{L}(\Theta, \boldsymbol{Y}) = \sum_{s=1}^{S} \alpha_T(s). \tag{6.53}$$

Using induction, it can also be shown that the matrix notation for the likelihood of a HMM is:

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\pi}_1 D_1(\Gamma D_2)(\Gamma D_3)\cdots(\Gamma D_T)\mathbf{1}' = \boldsymbol{\alpha}_T\mathbf{1}' \tag{6.54}$$

where $\mathbf{1}' = (1, 1, \ldots, 1)_{1\times S}$.

In practice, for a large observation sequence, the likelihood computation would result in very small values for forward probabilities – this could consequently cause computer overflow. A common way to overcome this problem is to calculate the log-likelihood rather than the likelihood function. Chib (1996) proposed a modification to the forward recursion procedure; to compute the log-likelihood by the following recursive formula:

$$\ln \tau_t = \ln \alpha'_{t-1} + M_t + \ln\Big(\sum_{s=1}^{S} \exp\Big\{\ln p_s(y_t) + \ln\Big[\sum_{s'=1}^{S} \frac{\alpha_{t-1}(s')}{\tau_{t-1}}\gamma(s', s)\Big] - M_t\Big\}\Big) \tag{6.55}$$

where

$$M_t = \max_s \ln\Big(p_s(y_t)\sum_{s'=1}^{S}\gamma(s', s)\frac{\alpha_{t-1}(s')}{\tau_{t-1}}\Big). \tag{6.56}$$

Equation (6.55) scales the forward probabilities $\alpha_t(s)$ at each step, so it suppresses the possibility of computer overflow.

6.3.3 Forward-Backward Recursion

The forward-backward (FB) recursion algorithm is basically a way to compute the likelihood function, developed by Baum et al. (1970). The FB recursion algorithm computes the likelihood function in two steps: forward recursion which accumulates information about the distribution of S_t as it moves down the hidden Markov chain; and the backward recursion which updates the distribution of S_t calculated in the forward step once information has been collected from all observed data. It needs to define the backward probabilities in addition to the forward probabilities. The backward probabilities $\beta_t(s)$ are referred to as the joint probability of the partial observation sequence after time t given the state event S_t at time t, that is

$$\beta_t(s) = \Pr\{\, Y_{t+1} = y_{t+1}, Y_{t+2} = y_{t+2}, \ldots, Y_T = y_T | S_t = s, \boldsymbol{\theta}\} \tag{6.57}$$

where $t = 1, 2, \ldots, T$ and $s = 1, 2, \ldots, S$. These probabilities are known as backward probabilities as they are calculated in a backward recursive method. Analogous to the forward probabilities we may represent backward probabilities by a vector $\boldsymbol{\beta}_t = [\beta_t(1), \beta_t(2), \ldots, \beta_t(S)]$. The last backward probability is set to

$$\boldsymbol{\beta}_T = [\, \beta_T(1), \beta_T(2), \ldots, \beta_T(S)\,] = [1, 1, \ldots, 1]_{1\times S}, \tag{6.58}$$

and simple calculation gives us the following recursive form for backward probabilities:

$$\boldsymbol{\beta}_{t+1} = \Gamma\, D_{t+1}\boldsymbol{\beta}_t, \tag{6.59}$$

or

$$\boldsymbol{\beta}_t = (\Gamma D_{t+1})(\Gamma D_{t+2})\cdots(\Gamma D_T)\mathbf{1}'. \tag{6.60}$$

Given the model parameters $\boldsymbol{\theta}$, the $T \times S$ matrices $A = [\,\alpha_t(s)\,]$ and $B = [\,\beta_t(s)\,]$ can be calculated in a recursive manner.

Forward-backward probabilities are also related to the likelihood of a HMM. It can be shown that for any $1 \leq t \leq T$ we have

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{s=1}^{S} \alpha_t(s)\beta_t(s) = \boldsymbol{\alpha}_t\boldsymbol{\beta}_t, \tag{6.61}$$

as $\alpha_t(s)$ accounts for the partial observation until time t and the $\beta_t(s)$ consider the remainder of the observation sequence. Hence, the forward-backward recursion can be seen as a way of profiling S out of $\mathcal{L}(\boldsymbol{\theta}, S)$, by plugging in $\widehat{S}$, or integrating S out of $\mathcal{L}(\boldsymbol{\theta}, S)$ for example in (6.61).

Forward-Backward probabilities provide a way to associate the optimal state sequence for a HMM given the observation sequence. There are several ways to choose the optimality. One possibility is to choose the most likely state value of $s \in \mathcal{S}$ for S_t. Hence, one needs to find the probability of being at state $S_t = s$ at time t given the observation sequence $\boldsymbol{Y}$.

$$\phi_t(s) = \Pr\{S_t = s \mid \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\theta}\} \tag{6.62}$$

Equation (6.62) can be written in terms of forward-backward probabilities:

$$\phi_t(s) = \frac{\alpha_t(s)\,\beta_t(s)}{\sum_{s=1}^{S}\alpha_t(s)\beta_t(s)}. \tag{6.63}$$

Note that $\sum_{s=1}^{S}\phi_t(s) = 1$ so $\phi_t(s)$ acts like a probability measure. One may produce the most likely sequence of states by

$$\widehat{S}_t = \underset{1\leq s\leq S}{\arg\max}\{\phi_t(s)\} \qquad t = 1, 2, \ldots, T. \tag{6.64}$$

6.3.4 EM Algorithm in HMM

Baum et al. (1970) proposed a recursive algorithm for parameter estimation in Markov chains and Welch (2003) developed the technique for HMMs. The Baum-Welch algorithm

can be considered as a generalised expectation-maximization (EM) algorithm. The EM algorithm gives a way to compute the maximum likelihood estimates of parameters in the statistical models where the model depends on unobserved latent variables. This algorithm starts by initialising values for $\boldsymbol{\theta}$ and the hidden states S_t, then it repeats two steps, expectation (E) and maximisation (M), as long as the set of parameters $\widehat{\boldsymbol{\theta}}$ has not converged. The E-step involves evaluating the log-likelihood using the current estimate of the hidden states, and M-step entails maximising the expected log-likelihood found on the E-step with respect to the parameters of the model. These estimated values for the parameters are used to determine the distribution of the hidden states in the next E-step. Having initialised the parameter set $\boldsymbol{\theta}$, the EM algorithm for HMM includes performing the following two steps alternatively:

E-Step: one estimates $u_t(s)$ and $v_t(s, s')$ given the current estimate of $\boldsymbol{\theta}$, by taking their conditional expectation:

$$\begin{aligned} \widehat{u}_t(s) &= E[u_t(s) \mid \widehat{\boldsymbol{\theta}}\,] \\ &= \Pr\{S_t = s \mid \boldsymbol{Y} = \boldsymbol{y}, \widehat{\boldsymbol{\theta}}\} \\ &= \widehat{\alpha}_t(s)\, \widehat{\beta}_t(s)/\mathcal{L}(\widehat{\boldsymbol{\theta}}) \end{aligned} \tag{6.65}$$

$$\begin{aligned} \widehat{v}_t(s, s') &= E[v_t(s, s') \mid \widehat{\boldsymbol{\theta}}\,] \\ &= \Pr\{S_{t-1} = s, S_t = s' \mid \boldsymbol{Y} = \boldsymbol{y}, \widehat{\boldsymbol{\theta}}\} \\ &= \widehat{\gamma}(s, s')\, \widehat{\alpha}_{t-1}(s)\, \widehat{p}_{s'}(y_t)\, \widehat{\beta}_t(s')/\mathcal{L}(\widehat{\boldsymbol{\theta}}) \end{aligned} \tag{6.66}$$

M-Step: Maximisation involves estimating new values for $\widehat{\boldsymbol{\theta}}$ by maximising the complete likelihood (6.11). This depends on the distribution of $p_s(y_t)$; the computation has been represented for the number of distributions in §6.3.1. We use the EM algorithm to integrate S out of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{S})$.

6.4 Mixtures of Hidden Markov Models

In this section, we illustrate the class of mixture models in which the components are the hidden Markov model, usually known as mixture of Hidden Markov Model (MixHHM). We extend the computations for HMMs for which the observed variable follows the nominal discrete distribution. Mixture distributions are used when it is assumed that the data are from one of a range of patterns. The idea of mixture models was first developed by Pearson (1894) for a mixture of two univariate Normal distributions. Mixture models

provide a flexible class of models for density estimation when classical models from the library of known distributions fail to provide a good fit for data. Mixture models have also been used as a model-based clustering approach by which observations can be clustered into different groups through a probabilistic framework (McLachlan and Basford, 1988). For a comprehensive review of mixture models see Titterington and Makov (1985) and McLachlan and Peel (2000).

The application of MixHMM has been found useful in different fields during the last decade. Qi et al. (2007) used the model to approximate the similarity of two pieces of music by computing the distance between the associated model, where in their application the music is treated as a time series data sequence. It has also been found that this model performs well for identifying groups of genes in gene expression time-courses (Schliep et al., 2004). Dias et al. (2010) introduced the application of MixHMM in finance by modelling Asian stock markets indexes. Modelling Internet browsing behaviour using MixHMM was first introduced by Ypma and Heskes (2002). They used the classical ML approach to estimate the parameters of the model. This model provides the practical advantage of clustering the users of the website, as well as soft categorisation of webpages of the website at the same time.

Suppose a set of realizations $\mathcal{Y} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_I\}$, where each $\boldsymbol{Y}_i = \{Y_t^i,\, t = 1, 2, \ldots, T_i\}$ is the sequence of observations for item i. In other words, $\mathcal{Y}$ consists of I sequences of observations from a HMM with K different sets of parameters $\boldsymbol{\Theta}_k = (\pi_k, \Gamma_k, \Lambda_k)$ for $k = 1, 2, \ldots, K$. Thus, the full parameter space consists of K sets of HMM parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_k,\, k = 1, 2, \ldots, K\}$. Throughout this chapter, $\boldsymbol{\theta}$ denotes the vector of parameters, consists of all parameters of the model. We let the random variable C_i represent the associated HMM to which the i-th item belongs. The ML estimate of all the parameters is derived by maximising the complete likelihood of the MixHMM. Now we introduce the likelihood function of the MixHMM.

6.4.1 Likelihood of MixHMM

Having assumed that I items are independent conditionally, the likelihood is the product of the likelihoods of each of the observed sequences.

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \Pr\{\boldsymbol{Y}_1 = \boldsymbol{y}_1, \boldsymbol{Y}_2 = \boldsymbol{y}_2, \ldots, \boldsymbol{Y}_I = \boldsymbol{y}_I \mid \boldsymbol{\theta}\} && (6.67)\\ &= \prod_{i=1}^{I} \Pr\{\boldsymbol{Y}_i = \boldsymbol{y}_i \mid \boldsymbol{\theta}\} && (6.68)\\ &= \prod_{i=1}^{I} \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_i). && (6.69)\end{aligned}$$

The likelihood for the i-th item is

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_i) = \sum_{k=1}^{K} \Pr\{\boldsymbol{Y}_i \;=\; \boldsymbol{y}_i \mid C_i = k, \boldsymbol{\theta}_k\} \Pr\{C_i = k \mid \boldsymbol{\theta}_k\}, \tag{6.70}$$

where the vector $\boldsymbol{\theta}_k$ consists of all parameters associated with the k-th HMM component. The *membership probability* is defined as the conditional probability that the i-th item belongs to the k-th HMM given the observations for the i-th item:

$$\omega_k^i = \Pr\{C_i = k \mid \boldsymbol{Y}_i, \boldsymbol{\theta}_k\} \tag{6.71}$$

where $\sum_k \omega_k^i = 1$. The membership probabilities form the conditional distribution of C_i given the sequence of observations. The unconditional distribution of C_i, $\Pr\{C_i = k\} = \omega_k$ delivers no information for the membership of the i-th item and will be denoted without the index of i. The unconditional membership probabilities, ω_k, are essentially the mixture proportions which are non-negative and sum to one over k. In practice, ω_k represents the relative frequency of occurrence for each group in the population. Using mixture coefficients ω_k the equation (6.70) can be expressed as follows:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_i) = \sum_{k=1}^{K} \mathcal{L}(\boldsymbol{\theta}_k|\boldsymbol{y}_i)\, \omega_k, \tag{6.72}$$

where $\mathcal{L}(\boldsymbol{\theta}_k|\boldsymbol{y}_i)$ is the likelihood for i-th observation when the class of model is known. The likelihood in the MixHMM needs to be represented based on the complete likelihood function which considers both the hidden state variables and observed sequence data. Thus, the conditional probability of the likelihood will be replaced by its complete likelihood function.

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_k|\boldsymbol{y}_i) &= \Pr\{\boldsymbol{Y}_i = \boldsymbol{y}_i \mid C_i = k, \boldsymbol{\theta}_k\}\\ &= \sum_{s^i \in \mathcal{S}^T} \Pr\{\boldsymbol{Y}_i = \boldsymbol{y}, \boldsymbol{S}_i = \boldsymbol{s}_i \mid C_i = k, \boldsymbol{\theta}_k\} && (6.73)\\ &= \sum_{s^i \in \mathcal{S}^T} \mathcal{L}(\boldsymbol{\theta}_k, \boldsymbol{S}_i|\boldsymbol{y}_i) && (6.74)\end{aligned}$$

Note that the parameter space for the conditional probability terms given $C_i = k$ is only limited to the Θ_k. The complete likelihood is similar to the (6.11) given in the previous section.

$$L(\boldsymbol{\theta}_k, \boldsymbol{S}_i \mid \boldsymbol{y}_i) = \pi_k(s_1^i)\gamma_k(s_1^i, s_2^i)\gamma_k(s_2^i, s_3^i)\cdots\gamma_k(s_{T_i-1}^i, s_{T_i}^i)\prod_{t=1}^{T_i} p_{s_t^i}^k(y_t^i). \tag{6.75}$$

6.4.2 Maximisation of the complete log-likelihood for MixHMM

The EM algorithm for the MixHMM does not appear to have been described or employed before. It can be implemented by generalization of the algorithm introduced for HMM. In this section we illustrate how the EM algorithm can be implemented for MixHMM. Now let

$$\begin{aligned}
w_k^i &= \begin{cases} 1 & \text{if } C_i = k \\ 0 & \text{if } C_i \neq k \end{cases} \\
u_t^{i,k}(s) &= \begin{cases} 1 & \text{if } S_t^i = s \text{ and } C_i = k \\ 0 & \text{if } S_t \neq s \end{cases} \\
v_t^{i,k}(s, s') &= \begin{cases} 1 & \text{if } S_{t-1} = s \text{ and } S_t = s' \text{ and } C_i = k \\ 0 & \text{if } S_{t-1} \neq s \text{ or } \quad S_t \neq s' \text{ and } C_i = k \end{cases}
\end{aligned} \tag{6.76}$$

then the logarithm of the complete likelihood can be written as the sum of three terms,

$$\begin{aligned}
\ell(\boldsymbol{\theta}, \boldsymbol{S}, \boldsymbol{C}) &= \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} u_1^{i,k}(s) \ln \pi_{1,k}(s) \\
&+ \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S}\sum_{s'=1}^{S} \left(\sum_{t=2}^{T} v_t^{i,k}(s, s') \right) \ln \gamma_k(s, s') \\
&+ \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S}\sum_{t=1}^{T} u_t^{i,k}(s) \ln p_s^k(y_t).
\end{aligned} \tag{6.77}$$

Theorem 6.8 *The first term of the complete log-likelihood function of MixHMM in* (6.77) *is maximised by the following initial state probabilities:*

$$\widehat{\pi}_{1,k}(s) = \frac{\sum_{i=1}^{I} w_k^i u_1^{i,k}(s)}{\sum_{i=1}^{I} w_k^i \sum_{s=1}^{L} u_1^{i,k}(s)}$$

Proof In order to maximise the first term of the complete likelihood (6.77), we consider

the constraint of $\sum_{s=1}^{S} \pi_{1,k}(s) = 1$. Defining the function F,

$$F = \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} u_1^{i,k}(s) \ln \pi_{1,k}(s) + \eta \left(1 - \sum_{s=1}^{S} \pi_{1,k}(s) \right), \tag{6.78}$$

where η is the Lagrange multiplier. The derivatives of F with respect to $\pi_{1,k}(s)$ for $s = 1, 2, \ldots, S$,

$$\frac{\partial F}{\partial \pi_{1,k}(s)} = \frac{\sum_{i=1}^{I} w_k^i u_1^{i,k}(s)}{\pi_{1,k}(s)} - \eta, \tag{6.79}$$

implies that $\eta = \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} u_{1,k}(s)$. Therefore,

$$\widehat{\pi}_{1,k}(s) = \frac{\sum_{i=1}^{I} w_k^i u_{1,k}(s)}{\sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} u_{1,k}(s)} \tag{6.80}$$

□

Theorem 6.9 *The second term of the complete log-likelihood function in* (6.77) *is maximised by the following transition probabilities:*

$$\widehat{\gamma}_k(s, s') = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t^{i,k}(s, s')}{\sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} u_t^{i,k}(s)}.$$

Proof For the second term of equation (6.77), we consider that there are S constraints, $\sum_{s'=1}^{S} \gamma_k(s, s') = 1$, corresponding to the rows of the transition matrix. So, we define the function G, .

$$G = \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} \sum_{s'=1}^{S} \left(\sum_{t=2}^{T} v_t^{i,k}(s, s') \right) \ln \gamma_k(s, s') + \sum_{s=1}^{S} \eta_s \left(1 - \sum_{s'=1}^{S} \gamma_k(s, s') \right), \tag{6.81}$$

where $\eta_1, \eta_2, \ldots, \eta_S$ are the Lagrange multipliers, then

$$\frac{\partial G}{\partial \gamma_k(s, s')} = -\eta_s + \frac{1}{\gamma_k(s, s')} \sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t^{i,k}(s, s'). \tag{6.82}$$

Hence, if we consider $-\eta_s \gamma(s, s') + \sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t(s, s') = 0$ then

$$\sum_{s'=1}^{S} \Big(-\eta_s \gamma(s, s') + \sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t(s, s') \Big) = 0. \tag{6.83}$$

Since $\sum_{s'=1}^{S} \gamma_1(s, s') = 1$ then $\eta_s = \sum_{s'=1}^{S} \sum_{t=2}^{T} v_t(s, s')$ so that

$$\widehat{\gamma}_k(s, s') = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t^{i,k}(s, s')}{\sum_{s'=1}^{S} \sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t^{i,k}(s, s')} \tag{6.84}$$

$$= \frac{\sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} v_t^{i,k}(s, s')}{\sum_{i=1}^{I} w_k^i \sum_{t=2}^{T} u_t^{i,k}(s)}. \tag{6.85}$$

□

Theorem 6.10 (Discrete nominal distribution)*: The third term of the complete log-likelihood function in* (6.77) *is maximised by the following value for the parameter of the discrete nominal distribution:*

$$\widehat{p}_s^k(y) = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1 \wedge Y_t = y}^{T} u_t^{i,k}(s)}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s)},$$

where $p_s^k(y)$ is the probability of observing y for the group of observation in the hidden state s and the k-th mixture component.

Proof The same procedure as described for HMM, provides us with the ML estimate of the parameters. In this case we first define

$$z_t^{i,k}(y,s) = \begin{cases} 1 & \text{if } Y_t^i = y \text{ and } S_t^i = s \text{ and } C_i = k \\ 0 & \text{if } Y_t \neq y \text{ or } \quad S_t \neq s \text{ and } C_i = k \end{cases} \tag{6.86}$$

where $y \in \mathcal{S}_y$ is the set of all possible values of Y_t. This way, the third part of the equation (6.77) can be rewritten as:

$$\sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{y \in \mathcal{S}_y} z_t^{i,k}(y,s) \ln p_s^k(y). \tag{6.87}$$

Considering the constraints over the parameter space, the function H needs to be maximised.

$$H = \sum_{i=1}^{I} w_k^i \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{y \in \mathcal{S}_y} z_t^{i,k}(y,s) \ln p_s^k(y) + \sum_{s=1}^{S} \eta_s \Big(1 - \sum_{y \in \mathcal{S}_y} p_s^k(y)\Big). \tag{6.88}$$

The derivative of H with respect to the parameters of the probability distribution over Y_t, the probability of the observation in this case, provides the ML estimate of the parameters.

$$\frac{\partial H}{\partial p_s^k(y)} = \sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} \frac{z_t^{i,k}(y,s)}{p_s^k(y)} - \eta_s. \tag{6.89}$$

The solution of the equation $-\eta_s p_s^k(y) + \sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} z_t^{i,k}(y,s) = 0$ gives us the values for the parameter space which maximise the function H. Note that $\sum_{y \in \mathcal{S}_y} p_s^k(y) = 1$, so summation over $\mathcal{S}_y$ gives us the η_s:

$$\eta_s = \sum_{i=1}^{I} w_k^i \sum_{y \in \mathcal{S}_y} \sum_{t=1}^{T} z_t^{i,k}(y,s) = \sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s). \tag{6.90}$$

so that

$$\widehat{p}_s^k(y) = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} z_t^{i,k}(y,s)}{\sum_{i=1}^{I} w_k^i \sum_{y \in \mathcal{S}_y} \sum_{t=1}^{T} w_t(y,s)} \tag{6.91}$$

$$= \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1 \wedge Y_t = y}^{T} u_t^{i,k}(s)}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s)}. \tag{6.92}$$

□

These probabilities can be summarized into a matrix notation $P_k = [\, p_s^k(y) \,]$ of dimension $S \times L$. So $\Lambda = \{P_1, P_2, \ldots, P_K\}$ is the set of parameters.

For the remainings distributions we do not write the proof, as it can be obtained very similar to the way we presented for HMM. That is, one needs to make the partial derivative of the third part of the log-likelihood equation (6.77) with respect to the set of parameters. The answer to the system of equation made by partial derivatives equal to zero, maximise the equation.

Theorem 6.11 (Poisson distribution)*: The third term of the complete log-likelihood function in* (6.77) *is maximised by the following value for the parameter of the Poisson distribution:*

$$\widehat{\theta}_s^k = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,t}(s) y_t}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,t}(s)},$$

where θ_s^k is the rate parameter of the Poisson distribution for the hidden state s and the k-th mixture component.

Theorem 6.12 (Exponential distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the parameter of the Exponential distribution:*

$$\widehat{\theta}_s^k = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s)}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s) y_t}.$$

where θ_s is the rate parameter of the Exponential distribution for the hidden state s and the k-th mixture component.

Theorem 6.13 (Binomial distribution)*: The third term of the complete log-likelihood function in* (6.13) *is maximised by the following value for the parameter of probability of success in the Binomial distribution, Where the number of experiments n_t is known:*

$$\widehat{\theta}_s^k = \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s) y_t}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s) n_t}.$$

where θ_s is the probability of success in the Binomial distribution for the hidden state s and the mixture component k.

Theorem 6.14 (Normal distribution)*: The third term of the complete log-likelihood function in* (6.77) *is maximised by the following value for the mean and variance parameters of the Normal distribution:*

$$\begin{aligned}
\widehat{\mu}_s^k &= \frac{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s) y_t}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s)} \\
\widehat{\sigma}_s^k &= \sqrt{\frac{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s)(y_t - \widehat{\mu}_s^k)^2}{\sum_{i=1}^{I} w_k^i \sum_{t=1}^{T} u_t^{i,k}(s)}}.
\end{aligned}$$

where μ_s^k and σ_s^k are respectively the mean and standard deviation parameters of the Normal distribution for the hidden state s and mixture component k.

The set of emission parameters for the Poisson, Exponential and the binomial can be represented in a $K \times S$ matrix of $\Lambda = [\, \theta_k^s \,]$, where cell (k, s) represent the parameter of the distribution for s-th hidden hidden states and k-th mixture components. Similarly for the Normal distribution, we can put the mean parameters into a matrix $M = [\, \mu_{k,l} \,]$ and variance parameters into $\Sigma = [\, \sigma_{k,l}^2 \,]$ both of dimension $K \times S$. The set of parameters is $\Lambda = \{M, \Sigma\}$.

For the rest of the chapter we illustrate the model when the Y_t^i follows the discrete nominal distribution, as the result will be used for the modelling web browsing behaviour in terms of visiting web pages. We assume the discrete nominal distribution for observing different pages of the website.

6.4.3 EM Algorithm for MixHMM

The maximisation of complete log-likelihood function for MixHMM enables us to find the ML estimates of the parameters of the model using EM algorithm. The EM algorithm for MixHMM be carried out by initialising $\boldsymbol{\theta}_k$ and repeating the following two steps until $\widehat{\boldsymbol{\theta}}$ has converged.

E-Step: estimate w_k^i, $u_t(s)$ and $v_t(s, s')$ given the current estimate of $\boldsymbol{\theta}_k$ by taking their conditional expectation:

$$\begin{aligned}
\widehat{u}_t^{i,k}(s) &= E[u_t(s) \mid \widehat{\boldsymbol{\theta}}_k] && (6.93)\\
&= \Pr\{S_t^i = s \mid \boldsymbol{Y}_i = \boldsymbol{y}_i, \widehat{\boldsymbol{\theta}}_k\} && (6.94)\\
&= \widehat{\alpha}_t^{i,k}(s)\, \widehat{\beta}_t^{i,k}(s) / \mathcal{L}(\widehat{\boldsymbol{\theta}}_k, \boldsymbol{S}_i \mid \boldsymbol{y}_i) && (6.95)
\end{aligned}$$

$$\begin{aligned}
\widehat{v}_t^{i,k}(s, s') &= E[v_t^{i,k}(s, s') \mid \widehat{\boldsymbol{\theta}}_k] && (6.96)\\
&= \Pr\{S_{t-1}^i = l, S_t^i = l' \mid \boldsymbol{Y}_i = \boldsymbol{y}_i, \widehat{\boldsymbol{\theta}}_k\} && (6.97)\\
&= \widehat{\gamma}^{i,k}(s, s')\, \widehat{\alpha}_{t-1}^{i,k}(s)\, \widehat{p}_{s'}^{i,k}(y_t)\, \widehat{\beta}_t^{i,k}(s') / \mathcal{L}(\widehat{\boldsymbol{\theta}}_k, \boldsymbol{s}_i \mid \boldsymbol{y}_i) && (6.98)
\end{aligned}$$

where $\alpha_t^{i,k}(s)$ and $\beta_t^{i,k}(s)$ are forward and backward probabilities respectively, given the i-th observation for k-th HMM. The expectation step also involves updating the membership

parameter:

$$\begin{aligned} \widehat{w}_k^i &= E[w_k^i] && (6.99)\\ &= \Pr\{C_i = k \mid \boldsymbol{Y}_i\} && (6.100)\\ &= \frac{\omega_k \Pr\{\boldsymbol{Y}_i \mid C_i = k, \Theta_k\}}{\sum_k \omega_k \Pr\{\boldsymbol{Y}_i \mid C_i = k, \Theta_k\}} && (6.101) \end{aligned}$$

M-Step: Estimate new values for $\widehat{\boldsymbol{\theta}}$ by maximising the complete likelihood $\mathcal{L}(\boldsymbol{\theta}_k, \boldsymbol{s}_i \mid \boldsymbol{y}_i)$. In other words, at this step the parameter space $\boldsymbol{\Theta}$ is updated. This step also includes updating the mixture probabilities ω_k by averaging membership probabilities over all items $i = 1, 2, \ldots, I$.

$$\widehat{\omega}_k = \frac{1}{I} \sum_{i=1}^{I} w_k^i. \tag{6.102}$$

For each Θ_k, this is a weighted version of the computation presented for HMMs, with respect to the mixture probabilities.

$$\widehat{\pi}_k(s) = \frac{\sum_{i=1}^{I} \omega_k^i \widehat{u}_1^{i,k}(s)}{\sum_{i=1}^{I} \omega_k^i \sum_{s=1}^{L} \widehat{u}_1^{i,k}(s)} \tag{6.103}$$

$$\widehat{\gamma}_k(s, s') = \frac{\sum_{i=1}^{I} \omega_k^i \sum_{t=2}^{T_i} \widehat{v}_t^{i,k}(s, s')}{\sum_{i=1}^{I} \omega_k^i \sum_{s=1}^{L} \sum_{t=2}^{T_i} \widehat{v}_t^{i,k}(s, s')} \tag{6.104}$$

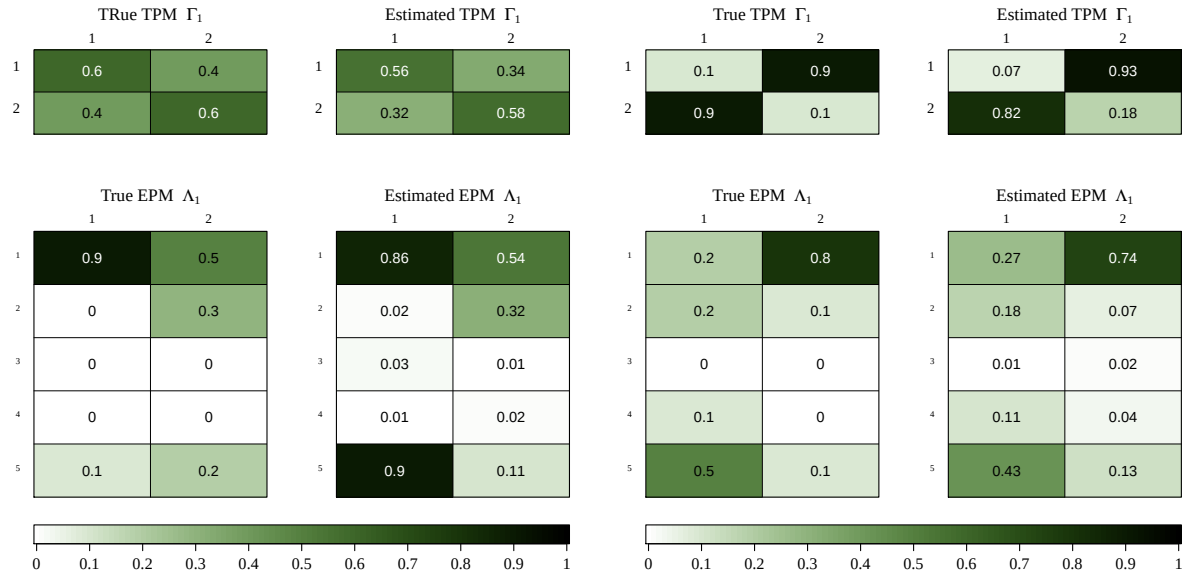
The maximisation of the third component of Λ depends on the distribution of $p_s^k(y)$. We have already extended these parameters in previous section for the MixHMM. After sufficient iteration of EM algorithm, it provides the ML estimate of the parameters in the model.

6.4.4 Simulation Study

We used the function `hmm` command of the `R` package `hmm.discnp` to compute the ML estimates of the parameters of HMM for the discrete nominal distribution by the EM algorithm. This function was modified to implement the EM algorithm for MixHMM. Our function is also able to run the *tied model*, where a common emission matrix is fit for all mixture components (see Appendix B, function `mixhmm`). It should be noted that, through correspondence with the package maintainer, Rolf Turner, this resulted in finding a bug in the codes regarding the mixture argument of the package.

We investigated the performance of the EM algorithm for MixHMM by generating an artificial data with several different settings. We present a simple example here which

Figure 6.3: *Graphical representation of true transition and emission matrices and the estimated values using EM algorithm (right)*



consists of 100 sequences of observations from HMM with two different set of parameters. The length of sequences varies by a average of 150. The components of HMM in the mixture model have 2 hidden states and emit 5 different values. The components are generated such that 40% of the sequences come from the first component and 60% from the second component respectively. Each sequence is labelled by $i = 1, 2$, showing the HMM by which the data has been generated. These labels help in our analysis to check whether the MixHMM is able to distinguish the cluster of sequences generated from the same HMM correctly. The length of the sequence is chosen by a discrete uniform distribution from 100 to 150. The transition and emission matrices for the HMM components are as follow:

$$\Gamma_1 = \begin{bmatrix} 0.60 & 0.40 \\ 0.40 & 0.60 \end{bmatrix} \qquad \Gamma_2 = \begin{bmatrix} 0.10 & 0.90 \\ 0.90 & 0.10 \end{bmatrix}$$

$$\Lambda_1 = \begin{bmatrix} 0.90 & 0.50 \\ 0.00 & 0.30 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.10 & 0.20 \end{bmatrix} \qquad \Lambda_2 = \begin{bmatrix} 0.20 & 0.80 \\ 0.20 & 0.10 \\ 0.00 & 0.30 \\ 0.10 & 0.00 \\ 0.50 & 0.10 \end{bmatrix}$$

Figure 6.3 shows the graphical representation of true transition and emission matrices and the estimated values using EM algorithm for MixHMM. A green colour spectrum is used

to represent the magnitude of transition and emission probabilities, from white for 0 to dark green for 1. The graph shows that the EM estimates of parameters are reasonably close to the true values, making us sure that there exists a good correspondence between learned and true parameter values. Similarly, it provides a very close estimation for initial probabilities, $\boldsymbol{\pi}_i$. For another example of MixHMM with discrete nominal observations for an artificial data set, as well as real data set application see Ypma and Heskes (2002). Although, the algorithm successfully gives us point estimates of the parameters, it does not provide any information about the precision of the estimates. Fuh and Hu (2007) showed how to use bootstrap to find a confidence interval for the parameters of a HMM, but with our best knowledge there has been no research in this area for the MixHMM.

The maximum likelihood approach to mixture models or HMM obtains point estimates of parameters by maximising the likelihood function. Extending the EM algorithm for more complicated models such as nested HMM or MixHMM, needs the analyst to develop the mathematical formula and also to produce the codes. Note that for high dimensions, apart from the computational difficulties associated with finding the local maximum of the likelihood surface, there usually exist several local maxima. This way, the ML gives different estimates for quantities of interest such as transition and emission probabilities depending on the initial plug-in values. Thus, it is necessary to choose between the local maxima where in many cases it is not easy to choose between them. Several local maxima may also cause a problem in the model selection phase, where we must choose the number of components as well as the number of hidden states/regimes. A standard method for choosing the number of components, K, applies the likelihood value of the fitted model, or a penalised version of the likelihood such as Akaike Information Criterion, known as AIC (Akaike, 1974). Hence, different values of the local maximum likelihood give different values for AIC. This way, likelihood-based criteria do not help in choosing a suitable value for the number of components. Due to the obstacles to the ML approach in both mixture models and HMMs we aim to use the Bayesian paradigm for MixHMM, where the MCMC approach serves as an alternative method by producing samples from the posterior distribution of the parameters.

6.5 Bayesian Inference of MixHMM

One of the main reasons for using sampling methods is that they enable an analyst to apply more complex structure models for which other techniques, if feasible, are highly complicated to implement (Puolamki and Kaski, 2009). In our case, an MCMC method enables us to implement MixHMMs without recursive EM algorithms, although recursive

methods can help for rapidly mixing in the MCMC algorithm. Bayesian inference for HMMs via Gibbs sampling has been developed by Albert and Chib (1993) and Robert et al. (1993) with the single-site updating of any hidden state. Chib (1996) proposed forward-filtering backward-sampling (FF-BS) to generate the sequence of the hidden states through the algorithm. We aim to extend the Bayesian inference to the MixHMM in the context of web usage data analysis, where a considerable number of observations is collected every day. In this section we illustrate the hierarchical structure of the proposed model and the assumptions on parameters of the model. We choose a uniform prior distribution over the parameters of the model. Updating the parameters is illustrated for both methods of direct Gibbs sampling and stochastic forward-backward recursion.

One of the specific advantages of using Bayesian MixHMM over the classical maximum likelihood approach is related to the problem of zero probabilities. In the MixHMM, this problem occurs when there are no transitions observed between states or no observation of $y \in \mathcal{S}_y$ when the model is in the hidden state $s \in \mathcal{S}$ in the sequence of observations. In such a case, the ML estimate of the transition probability in each mixture component will be zero. To deal with this problem in the Bayesian context, one can consider such a prior for transition parameters to enforce the belief that all transitions are possible (Cadez et al., 2000b). Bayesian MixHMM also enables the analyst to compute the probability intervals of interest, for example the probability interval for a session containing an online purchase or visiting the contact page. Note that, prior knowledge can be applied in terms of a prior distribution over the parameters of the model. The model can also be updated, by sequential Bayesian analysis, using recent data collected from web server logs.

In the Bayesian approach, given the data set $\boldsymbol{y}$ one aims to obtain the posterior distribution of the vector of parameters $\boldsymbol{\theta}$:

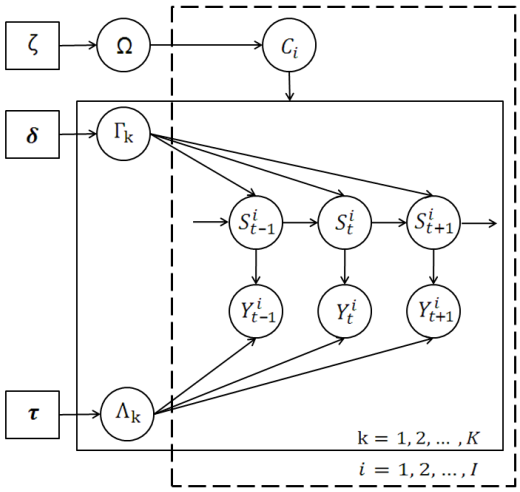
$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}) \tag{6.105}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood function and $p(\boldsymbol{\theta})$ is the prior distribution of the parameters. In some applications an analyst is also interested in obtaining the joint distribution of the state and the parameters $p(\boldsymbol{s}, \boldsymbol{\theta}|\boldsymbol{y})$. In the next section, we define the complete hierarchical model for MixHMM and the set of parameters in the model. The prior distribution over parameters and hidden variable, as well as observed variables will be discussed.

6.5.1 Complete Hierarchical Model

Consider a dataset consisting of I different sequences of output observations, where each follows the HMM with a discrete nominal distribution for sequences of observations. Fig-

Figure 6.4: *The graphical representation of the joint distribution of MixHMM.*



ure 6.4 depicts the DAG of the hierarchical MixHMM. Circles represent the unknowns or observed quantities, including variables and parameters. Square boxes represent fixed hyper-parameters. The part of the DAG placed in the dashed-line box shows that for $i = 1, 2, \ldots, I$ individuals this model is duplicated. The components of MixHMM have been surrounded by a solid-line box for $k = 1, 2, \ldots, K$, where Γ_k and Λ_k are transition and emission matrices for the k-th HMM respectively. Hyper-parameters in a hierarchical structure helps us to get weakly informative priors for the parameters, and to make only minimal assumptions on the data.

The quantity C_i is the hidden variable which we refer to as the membership variable, since it takes values k in $\{1, 2, \ldots, K\}$, showing that item i belongs to the model k. Hence, C_i has a discrete nominal distribution with unknown probability parameters, known as membership probabilities. We represent all membership probabilities in a matrix form $\Omega = [\, \omega^i_k \,]_{T \times K}$ and call Ω the membership matrix later on, as the i-th row of the k-th column of the membership probability matrix, ω^i_k is the probability that the item i follows the k-th HMM. The vectors $\boldsymbol{\zeta}$, $\boldsymbol{\delta}$, and $\boldsymbol{\tau}$ are fixed hyper-parameters in the hierarchical structure of the model. The joint distribution of all the variables conditional on fixed hyper-parameters may be written as:

$$p(\Omega, \Gamma, \Lambda, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{c} | \boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\zeta}) \propto p(\Omega|\boldsymbol{\zeta}) p(\boldsymbol{c}|\Omega) p(\Gamma|\boldsymbol{\delta}, \boldsymbol{c}) p(\Lambda|\boldsymbol{\tau}, \boldsymbol{c}) p(\boldsymbol{s}|\boldsymbol{c}, \Gamma) p(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{c}, \Lambda). \quad (6.106)$$

In the next section, we explain the prior distributions of Γ_k, Λ_k; the distribution of hidden membership variables, C_i, and hidden state variables, S^i_t; and the distribution over observed variable Y^i_t.

6.5.2 Prior Information

The posterior distribution of the parameters, given the observation sequences, depends on the prior distribution assumed for the parameters. In our discrete nominal case (and also multinomial distribution), an often-used candidate prior is the Dirichlet distribution. The Dirichlet distribution, denoted by $\mathcal{D}(\boldsymbol{\delta})$, is a continuous distribution function with the following density function for the vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_K)$:

$$f(\boldsymbol{y}; \boldsymbol{\delta}) = \frac{\Gamma(\sum_{i=1}^{K} \delta_i)}{\prod_{i=1}^{K} \Gamma(\delta_i)} \prod_{i=1}^{K} y_i^{\delta_i - 1} \tag{6.107}$$

subject to $\sum_i^K y_i = 1$, $0 < y_i < 1$, and $\delta_i > 0$. Note that the $\Gamma(\,\cdot\,)$ in 6.107 denotes the well-known gamma function, and $\boldsymbol{\delta}$ is a vector of positive real numbers of length K. The Dirichlet distribution is the multivariate generalization of the Beta distribution. It is a conjugate prior of the nominal discrete distribution and the Multinomial distribution.

We consider the product of independent Dirichlet distribution over the rows of the transition matrix Γ, as a prior for the transition probabilities.

$$\boldsymbol{\gamma}_s \sim \mathcal{D}(\delta_{s,1}^k\ ,\ \delta_{s,2}^k\ ,\ \ldots\ ,\ \delta_{s,S}^k) \tag{6.108}$$

where $\boldsymbol{\delta}_s^k = (\delta_{s,1}, \delta_{s,2}, \ldots, \delta_{s,S})$ is the hyper-parameter vector for the prior distribution. A vector of $\boldsymbol{\delta}_s = \mathbf{1}_S$ serves as a uniform prior for the rows of transition matrix on the simplex of dimension S, where $\mathbf{1}_k$ denotes a vector of 1's of dimension k. The Jeffreys' priors for the Multinomial distribution provides a non-informative prior by setting $\boldsymbol{\delta}_s = \mathbf{1}_S/S$ (Kass and Wasserman, 1996). For the emission matrix Λ, we assume the product of independent Dirichlet distributions over columns. The hyper-parameter $\boldsymbol{\tau} = \mathbf{1}_L$ gives a uniform prior, and $\boldsymbol{\tau} = \mathbf{1}_L/L$ is the Jeffreys' prior.

$$\boldsymbol{\lambda}_s \sim \mathcal{D}(\tau_{1,s}^k\ ,\ \tau_{2,s}^k\ ,\ \ldots\ ,\ \tau_{L,s}^k). \tag{6.109}$$

We denote $\boldsymbol{\tau}_s^k = (\tau_{1,s}, \tau_{2,s}, \ldots, \tau_{L,s})$ as the hyper-parameter vector for the prior distribution of Λ. We postulate the Dirichlet distribution over weight parameters $\boldsymbol{\omega}$.

$$\boldsymbol{\omega} \sim \mathcal{D}(\zeta_1\ ,\ \zeta_2\ ,\ \ldots\ ,\ \zeta_K). \tag{6.110}$$

The hyper-parameter vector for the prior distribution of $\boldsymbol{\omega}$ is denoted by $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \ldots, \zeta_K)$. By setting the parameter vector to $\boldsymbol{\zeta} = \mathbf{1}$ one assigns equally likely probabilities to the mixture proportion parameters.

6.5.3 Gibbs Sampler for MixHMMs

Gibbs sampling is a simulation method based on Markov chain theory in which one takes samples from conditional distributions instead of directly sampling from an original distribution. This way, it generates a Markov chain with a stationary distribution which is not independent and identically distributed. However, the extension of the law of large numbers guarantees for an approximation of any posterior quantity of interest based upon this sample (Geman and Geman, 1984; Tanner, 1991; G. and George, 1992; Smith and Roberts, 1993).

Using the Gibbs sampler for simulating samples from the parameters of the MixHMM involves three steps: (1) sampling from the membership state variable given the observation sequence and parameters (2) sampling from the hidden states variables given the observation sequence and parameter; (3) sampling from the distribution of the parameters given hidden states, membership states, and observation sequence. This alternate sampling produces a sequence of N triplets of parameters, hidden state, and membership states.

$$\{(\Theta, \boldsymbol{S}, \boldsymbol{C})_n, \quad n = 1, 2, \ldots, N\} \tag{6.111}$$

In order to generate realizations from the posterior joint distribution of the parameters, we alternate the following moves all through Gibbs sampling:

1. **updating the transition matrices $\boldsymbol{\Gamma}$:** The s-th row of transition matrix Γ_k is sampled from the following Dirichlet distribution:

$$\boldsymbol{\gamma}_s \sim \mathcal{D}(\delta^k_{s,1} + a_1 \, , \, \delta^k_{s,2} + a_2 \, , \, \ldots \, , \, \delta^k_{s,S} + a_S) \, , \tag{6.112}$$

   where $a_{s'} = \sum_{i=1}^{I} \sum_{t=2}^{T_i} v_t^{i,k}(s, s')$ is the number of transitions from component s to component s' over all items assigned into the class of k (Robert et al., 1993).

2. **updating the emission probabilities $\boldsymbol{\Lambda}$:** the s-th row of emission matrix Λ_k is sampled from the following Dirichlet distribution:

$$\boldsymbol{\lambda}_s \sim \mathcal{D}(\tau^k_{1,s} + b_1 \, , \, \tau^k_{2,s} + b_2 \, , \, \ldots \, , \, \tau^k_{L,s} + b_L) \tag{6.113}$$

   where $b_l = \sum_{i=1}^{I} \sum_{t=1 \wedge y_t = y(l)}^{T_i} u_t^{i,k}(s)$ counts how many times the value $y(l)$, the l-th element of the $\mathcal{S}_y$, appears in the component s of the class k.

3. **updating the membership variables $\boldsymbol{C}_i$:** The membership of each item is sampled directly from a discrete nominal distribution, where the parameters are membership probabilities of the item i, $(\omega_1^i, \omega_2^i, \ldots, \omega_K^i)$ given in the i-th row of the membership matrix Ω.

$$C_i \sim \mathcal{DN}(\omega_1^i\ ,\ \omega_2^i\ ,\ \ldots\ ,\ \omega_K^i). \tag{6.114}$$

The membership probabilities ω_k^i are computed at each iteration:

$$\omega_k^i \propto \pi_{c_i}(s_1^i)\gamma_{c_i}(s_1^i, s_2^i)\gamma_{c_i}(s_2^i, s_3^i)\cdots\gamma_{c_i}(s_{T-1}^i, s_T^i)\prod_{t=1}^{T_i} p_{s_t^i}(y_t^i)\,\omega_{c_i} \tag{6.115}$$

with the constraint of $\sum_{k=1}^{K}\omega_k^i = 1$.

4. **updating the hidden state variable $\boldsymbol{S}_t$**: A direct Gibbs sampling draws each hidden state variable S_t^i from its full conditional distribution using the following distribution:

$$\Pr\{S_t^i = s \mid \boldsymbol{S}_{-t}^i, \boldsymbol{y}, \Gamma\} \propto \gamma_{c_i}(s_{t-1}, s)\gamma_{c_i}(s, s_{t+1})p_{s_t}(y_t) \qquad t = 1, 2, \ldots, T. \tag{6.116}$$

where the $\boldsymbol{S}_{-t}^i$ denotes all state variables of the i-th individual, excluding the state at time t (Robert et al., 1993; Robert and Titterington, 1998). Scott (2002) introduced sampling from $p(\boldsymbol{s} \mid \boldsymbol{y}, \Gamma)$ using a stochastic version of the forward-backward (FB) recursion algorithm. This method gives a faster mixing algorithm, as fewer components are introduced into the Gibbs sampler. The stochastic FB algorithm modifies (6.116) by using a stochastic version of the forward-backward recursions to sample S_t directly from $\Pr(\boldsymbol{S}\,|\,\boldsymbol{\theta}, \boldsymbol{Y})$. The forward recursion step involves producing $A_2, A_3, \ldots, A_T$, as introduced in §6.3.3, and the stochastic backward recursion draws S_T from $\Pr(S_T|\boldsymbol{\theta}, \boldsymbol{Y})$ for $t = T-1, T-2, \ldots, 1$ from the distribution proportional to the column S_{t+1} of A_{t+1} respectively. In our analysis we use the stochastic FB sampling extended for MixHMM by conditioning the algorithm over the membership variable C_i.

5. **updating the mixture proportion parameters $\boldsymbol{\omega}$:** The $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_K)$ is sampled from the Dirichlet distribution:

$$\boldsymbol{\omega} \sim \mathcal{D}(\zeta_1 + d_1\ ,\ \zeta_2 + d_2\ ,\ \ldots\ ,\ \zeta_K + d_K) \tag{6.117}$$

where $d_k = \sum_{i=1}^{I} w^i(k)$ is the number of items/individuals over all items assigned into the class of k.

6.5.4 Model selection

When there exists no information about the dimensions of the model, it is necessary to select a model with respect to the number of hidden state, S, and the number of components, K. Traditionally, the likelihood function can help when choosing a model which better describes the data. In the likelihood approach, one may run the posterior sampler for a variety of models with different combinations of the $(S = s, K = k)$. It is always expected that the log-likelihood will be increased by increasing the number of the parameters in the model. An insignificant increase of the log-likelihood in comparison with a model with fewer parameters shows that higher dimension does not increase the ability of the model to describe the patterns observed in the data. Based on the parsimony principle of statistical modelling, a good model with fewer parameters is preferred. In the Bayesian approach where we have a distribution over the log-likelihood, it is common to sketch the box-plot of the log-likelihood of different models in the same plot to help make judgements (Scott and Hann, 2007). A penalised version of log-likelihood such as the Bayesian information criterion (BIC) can also be used to avoid the over-fitting problem. The BIC is given as:

$$\text{BIC} = -2\,\ell(\boldsymbol{\theta}) + p\ln(n), \tag{6.118}$$

where p is the number of free parameters to be estimated, and n the number of observations, or equivalently, the sample size. Similarly, a box-plot of the BIC measure in Bayesian analysis of the MixHMM can provide a comparison framework to choose the model with smallest BIC. The total number of parameters in a untied MixHMM need to be estimated is:

$$K \times (S \times S) + K \times (L \times S) + K, \tag{6.119}$$

including K transition matrices Γ_k with $S \times S$ parameter, K emission matrix Λ_k with $L \times S$ parameter, and an mixture proportion vector $\boldsymbol{\omega}$ of size K. The number of parameters for the tied MixHMM is reduced by considering a common emission matrix:

$$K \times (S \times S) + 1 \times (L \times S) + K. \tag{6.120}$$

Ypma and Heskes (2002) use a cross-validation method by partitioning a sample of the data into complementary subsets: training and test/validation data sets. The training data set is used to estimate the parameters of the model. Having the estimated parameters, the test data set is used to check if the model is good enough. This approach is mainly aimed at assessing how accurate the predictive model is in practice. They used the following score for out-of-sample test.

$$\text{score}(\boldsymbol{Y}_{test}) = -\frac{\sum_{i=1}^{I} \ln \sum_{k=1}^{K} \omega_k P(Y_{test} \mid C_i = k, \boldsymbol{\theta})}{N_{test}} \tag{6.121}$$

Another possibility for making inference about the unknown number of hidden states and number of mixture components might be the use of the so-called *reversible jump* algorithm which allows for the changing dimension of the parameter space. The idea of reversible jump Markov chain Monte Carlo (RJ-MCMC) was first introduced by Green (1995) and Richardson and Green (1997). Robert and Titterington (2000) showed how the RJ-MCMC techniques can be used to estimate the parameters of a HMM model, as well as the number of regimes for a mixture of normal distributions, and Castellano and Scaccia (2007) extended it to the mixture of normal distribution with different means and variances under each regime. To our best knowledge, RJ-MCMC has not yet been developed for MixHMM.

RJ-MCMC can serve to make an inference about the dimensionality of the model, but in practice, in particular for models with high complexity, this might be very difficult. For example consider the MixHMM for nominal discrete observations and its application for modelling Internet browsing behaviour. First, transformation from the parameter vectors of one model to another requires the computation of the determinant of the Jacobian matrix. This for such a high dimensional model, if not impossible, might be very difficult. Also, having a large sample size, as we have in the clickstream data, can significantly increase the time of running the RJ-MCMC, whilst it seems more efficient to let the Gibbs sampler be run over different machines for different models and use the results to make a comparison of the models.

6.6 Some issues in implementing the Gibbs sampler

6.6.1 Slow mixing

A common problem of using MCMC methods is so called *slow mixing*, when values $\theta^{(t)}$ generated at iteration t is look a lot like the $\theta^{(t+1)}$ and this similarity continues for all iterations, or in technical term there is a strong autocorrelation between the samples. This way, the contribution of each additional Gibbs sample to the quality of inferences about the posterior density is small. Hence, the sampler requires extremely long run of the MCMC algorithm from a slow-mixing MCMC algorithm to reach a sample size that adequately represents the posterior distribution, which might be practically not feasible. A simple solution for slow mixing is to retain every m-th MCMC iterate, where m is known as the *thinning interval*. The thinning interval is selected so that the samples are apart enough to approximate an independence sampler, *thinned* sequence of generated samples are expected to have a low autocorrelation. In addition to reducing autocorrelation,

thinning the sequence of samples also helps save computer memory and time (Jackman, 2009).

6.6.2 Label-Switching in MixHMM

When the Bayesian approach is applied to mixture models, one might face the so-called label-switching problem (Richardson and Green, 1997). This problem arises mainly because of the invariance of the likelihood with respect to the permutations of the component labels in the mixture model, or regimes in HMM (Redner and Walker, 1984). Similarly in Bayesian analysis, the posterior is symmetric when the prior distribution is the same for equivalent parameters in different components/regimes. Consequently, the Gibbs Sampler might mix up the samples component labels when label-switching occurs. So there is an arbitrariness in reporting the order of labels. In this respect, the sampler encounters the highly symmetric multi-modal posterior distribution. Similarly, the problem of label switching might occur in the HMM, as multiple ways of labelling the regimes (hidden states) can alternate during the MCMC iterations and fail to identify the HMMs parameters in model fitting.

Given a mixture model with K components, there are $K!$ symmetric modes of the posterior distribution. If the Gibbs sampler can thoroughly and evenly travel all the $K!$ symmetric modes, the posterior expectation for each component parameter should be identical. However, as K increases, $K!$ is very large and the sampler may fail to thoroughly and evenly explore the distribution surface. In this case, an unbalanced label-switch causes a multi-modal distribution surface for the posterior distribution. This feature provides a diagnostic to check label-switching (Jasra et al., 2005). One can also discover the problem of label-switching through overlaid trace-plots for equivalent parameters of different components of the mixture model. Unbalanced label-switching also cause a very poor estimate in estimating the marginal density from the Gibbs sampler such that the results might be very different from different runs (Fruhwirth-Schnatter, 2001). Note that the sampler might perform well in picking out the parameter from the data and label-switching may not occur at all. The unbalanced structure of the MixHMM does not let the label-switching occurs for components, But it needs to be checked for hidden state for each component. Thus we take advantage of trace plots to find out if the label-switching is taking place.

However, there is no guarantee that label-switching will not happen if the sampler is run with more iterations (Jasra et al., 2005). There are situations of extreme unbalanced label-switching. In this case, every component maintains its own mode in the long run

and the sampler does not observe any label-switching. The unbalanced label-switch is more likely to happen when the dimension of the model increases (Sperrin et al., 2010).

6.7 Simulation Study

Data generation

In this section we illustrate how MixHMM works for an artificial data set. We generate 200 sequences of observations by MixHMM, consisting of three components of HMM. The components of HMM in the mixture model have 3 hidden states and emit 5 different values. The components are generated such that 30% of the sequences are for the first component and 40% and 30% for the second and third component respectively. This way, the second component is 30% more likely to occur compared to the first and third component. Each sequence is labelled as $i = 1, 2, 3$, showing the HMM by which the data has been generated. These labels help in our analysis to check whether the MixHMM is able to distinguish the cluster of sequences generated from the same HMM correctly. The length of the sequence is chosen by a discrete uniform distribution from 2 to 100. The transition and emission matrices for HMM components are as follows:

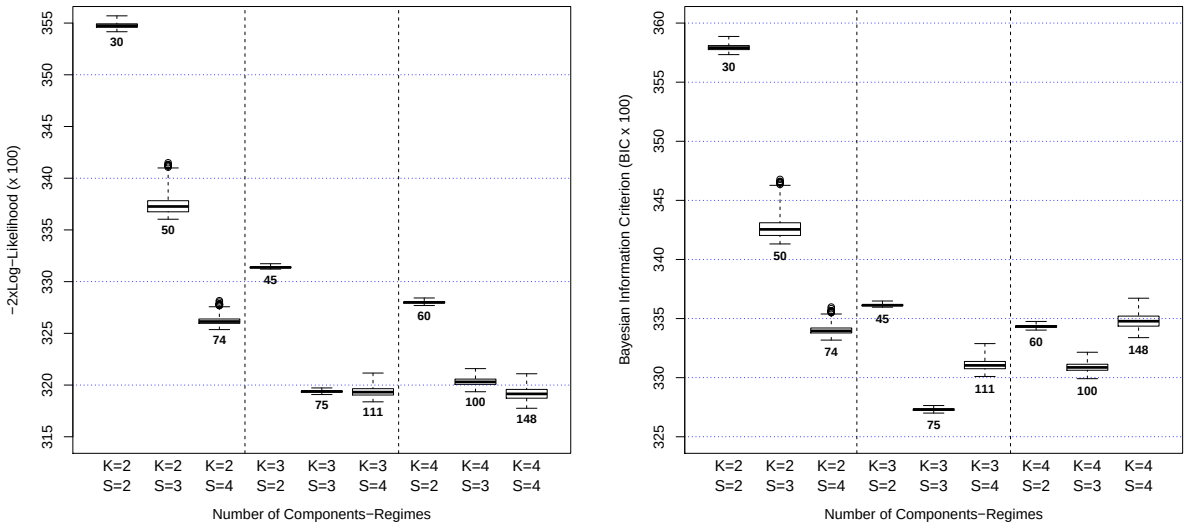
$$\Gamma_1 = \begin{bmatrix} 0.80 & 0.10 & 0.10 \\ 0.10 & 0.80 & 0.10 \\ 0.00 & 0.20 & 0.80 \end{bmatrix} \quad \Gamma_2 = \begin{bmatrix} 0.90 & 0.10 & 0.00 \\ 0.20 & 0.70 & 0.10 \\ 0.10 & 0.00 & 0.90 \end{bmatrix} \quad \Gamma_3 = \begin{bmatrix} 0.20 & 0.80 & 0.00 \\ 0.00 & 0.30 & 0.70 \\ 0.60 & 0.20 & 0.20 \end{bmatrix}$$

$$\Lambda_1 = \begin{bmatrix} 0.90 & 0.50 & 0.30 \\ 0.00 & 0.00 & 0.10 \\ 0.00 & 0.00 & 0.10 \\ 0.00 & 0.00 & 0.20 \\ 0.10 & 0.50 & 0.30 \end{bmatrix} \quad \Lambda_2 = \begin{bmatrix} 0.10 & 0.30 & 0.20 \\ 0.85 & 0.30 & 0.20 \\ 0.05 & 0.40 & 0.20 \\ 0.00 & 0.00 & 0.20 \\ 0.00 & 0.00 & 0.20 \end{bmatrix} \quad \Lambda_3 = \begin{bmatrix} 0.00 & 0.00 & 0.10 \\ 0.00 & 0.00 & 0.00 \\ 0.10 & 0.10 & 0.80 \\ 0.10 & 0.80 & 0.00 \\ 0.80 & 0.10 & 0.10 \end{bmatrix}$$

Model selection

Model selection in MixHMM involves choosing the number of hidden states and the number of components. We ran the posterior sampler for models with 2 to 4 mixture components, where for each one the number of hidden states varies between 2 to 4. The samples from the posterior of the log-likelihood $\ell(\boldsymbol{\theta})$ for each model are recorded, based on 30,000 samples with the first 10,000 samples discarded as burn-in, and with a thinning interval of 10. Figure 6.5 (left) shows the box-plots representing the posterior distribution of $-2 \times \ell(\boldsymbol{\theta})$, where for each model with K mixture components the value of $-2 \times \ell(\boldsymbol{\theta})$

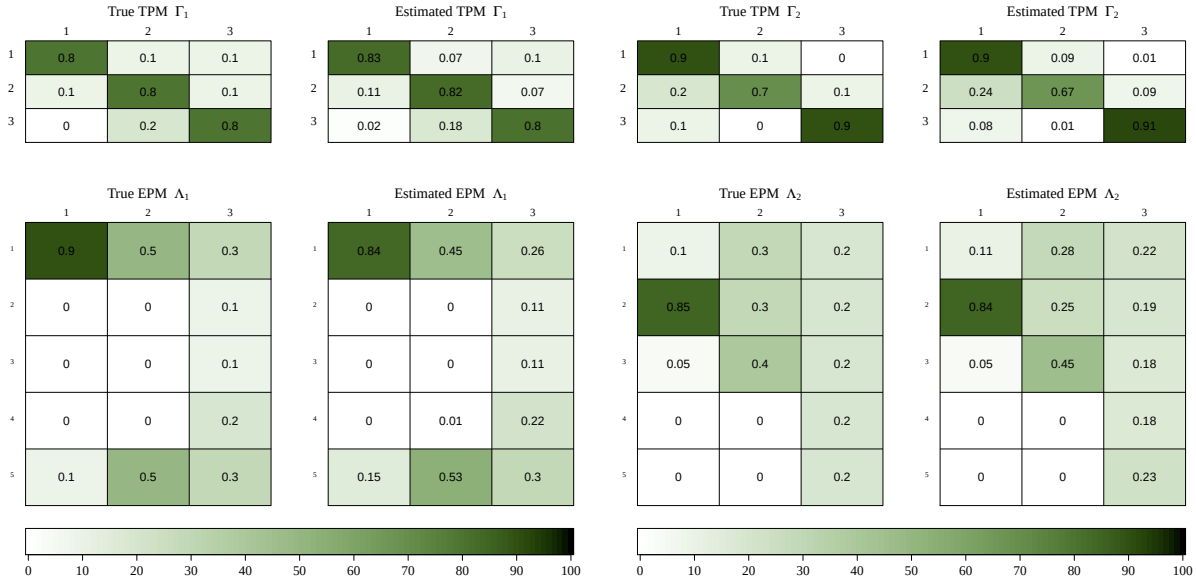
Figure 6.5: *The boxplot of the $-2\times$Log-likelihood (left) and BIC (right) for different models in terms of the number of mixture components K and the number of hidden states/regimes S. The number shown under each box-plot represents the number of parameters of the model*



decreases until $S = 3$, but little reduction is seen by increasing to $S = 4$. Figure 6.5 (right) shows the box-plots of the posterior distribution of the BIC for MixHMMs fitted varying the number of clusters K and the number of hidden states S. The smallest range of values has been produced for the model of $K = 3$ and $S = 3$. The result shows that BIC successfully helped to find the true number of hidden states and clusters. The remainder of this section describes the results received for models with $K = 3$ and $S = 3$.

After selecting the number of components and the number of hidden variables, the model will be run again for the selected model. We run the Gibbs sampler with 60,000 iterations, first 10,000 samples burn-in, and thinning interval of 10, we take a 5000 sample from the posterior of the parameters in MixHMM with $K = 3 - S = 3$. We will later see that the burn-in can help to take an uncorrelated sample from the posterior through Gibbs sampler. Figure 6.6 shows the graphical representation of true transition and emission matrices and the average over the posterior samples of transition and mission matrices using Gibbs samples. We only show the parameters of the first two components, where the third component shows a similar pattern to the first two components. A green colour spectrum is used to represent the magnitude of transition and emission probabilities, from white for 0 to dark green for 1. The graph shows that the average of the posterior samples is reasonably close to the true values, making us sure that there exists a good

Figure 6.6: *Graphical representation of true transition and emission matrices (left), and the average of estimated values transition and mission matrices using EM algorithm (right)*



correspondence between learned and true parameter values.

Figure 6.7 (left) displays the trace-plot for the Gibbs samples of the log-likelihood (in hundreds) of the model with $K = 3$ and $S = 3$. The trace-plot helps to check if the Gibbs sampler has eventually reached a stationary distribution, usually referred to as *convergence* of the sampler. The caterpillar shape of the trace-plot confirms that the samples of the conditional distributions of the parameters have reached the stationary status. Figure 6.7 (right) depicts the autocorrelation plot, known as the *ACF-plot*, of the likelihood samples. This plot shows that the thinning, have provide an uncorrelated sample of log-likelihood.

Figure 6.8 shows the trace-plot of the posterior sample of transition probabilities; each component is represented by a different colour. The paths reveal no evidence of a problem of label-switching of hidden states (Celeux et al., 2000; Stephens, 2000). This can be noticed by either swapping chains in the plot (changing colours) or the appearance of multiple modes. Figure 6.9 shows the MCMC sample paths of emission probabilities for the different components. similarly, it shows the trace plot of emission probabilities, where each plot contains emission probabilities for all mixture components with different colours. Both graphs show that no label-switching, either for mixture components or hidden states, has been placed. If the label switching happens we expect to see that the colour of a sequence swap between different ones.

Figure 6.7: *The trace-plot of log-likelihood values of the function at each iteration (left) and the ACF plot of values of the log-likelihood (right).*

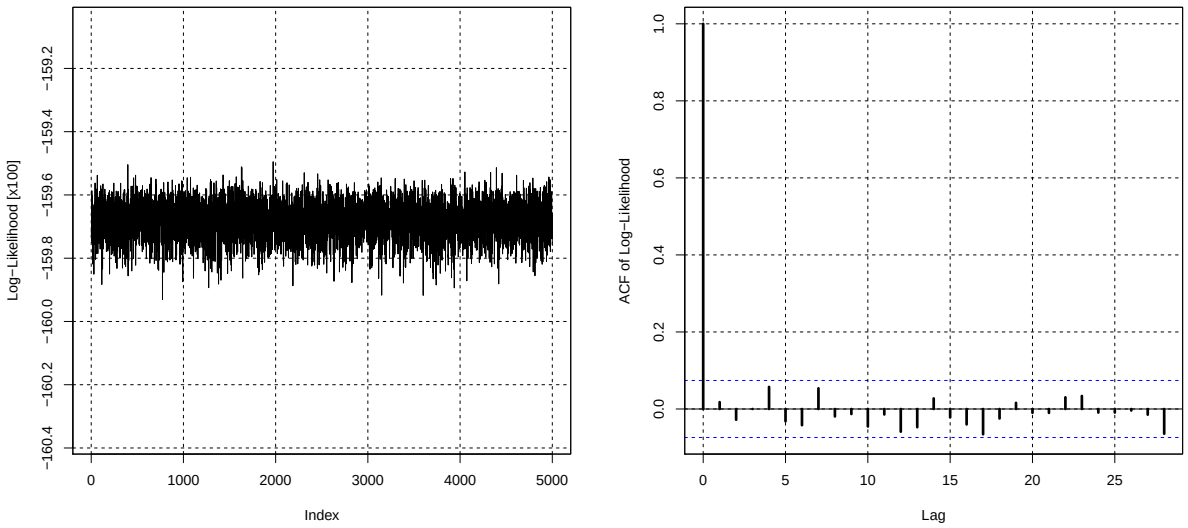


Figure **??** displays the trace-plot of Gibbs samples of transition probabilities $[\gamma_k(s, s')]$ for both classes. The plot does not show any evidence of non-stationary status after burn-in samples. Hence, we may use the samples of the conditional distributions of the parameters to summarize the posterior distribution of transition probabilities. This trace-plot for transition probabilities are also used as a diagnostic tool to check if there exists any label switching problem in mixture models and HMMs (Celeux et al., 2000). Similarly, Figure **??** displays the trace plots of the emission parameters. The shape of observation in the tace-plot gives us enough evidence for having a stationary sample from the posterior distribution of emission probabilities.

We also look at the ACF-plot of the parameters. There is a large numebr of parameters in the our MixHMM, so we just display the ACF-plot of the transition probabilities in Figure 6.11 for both mixture components. We observe that after thinning and using forward-backward recursion, it remains very low level of autocorrelation between the Gibbs samples.

In order to investigate whether the model is able to learn correct labelling/clustering of states, we observe the average of the Gibbs samples drawn for the elements of the membership matrix Ω. The results reveal that the model successfully assigns all individuals/items to the true clusters. We also check whether a misspecified model is able to perform labelling correctly when the number of components in the data is smaller than

Figure 6.8: *The trace-plot of transition probabilities.*

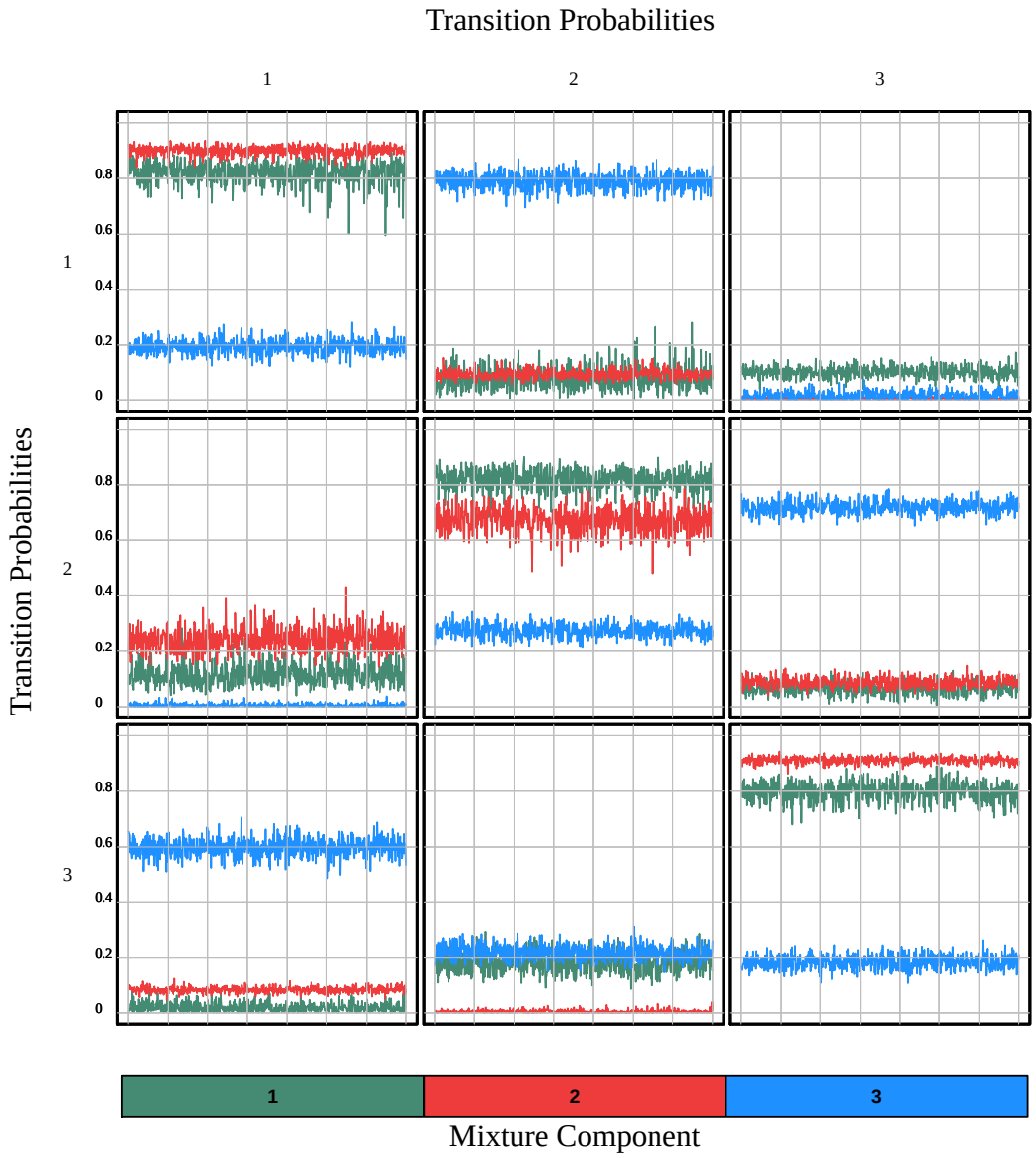


Figure 6.9: *The trace-plot of emission probabilities to investigate label switching between hidden states.*

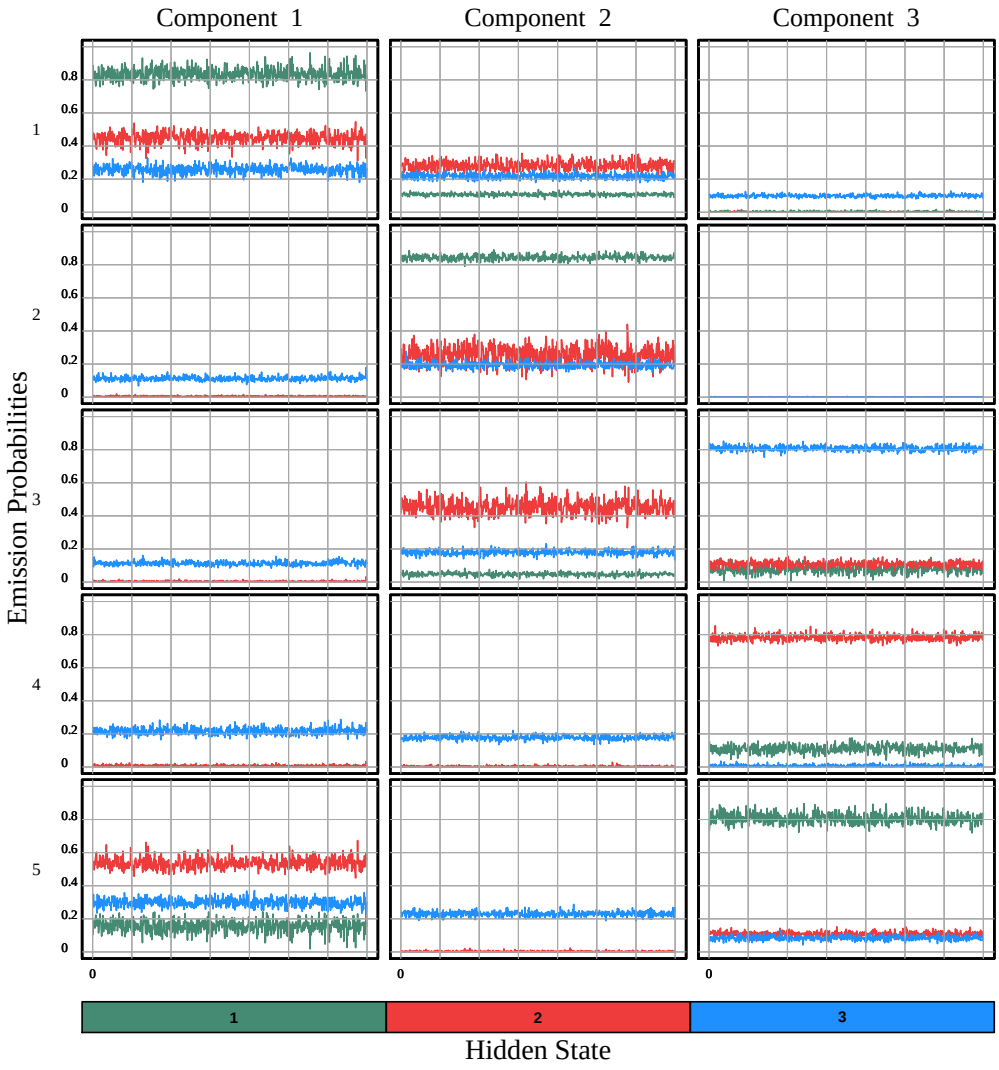


Figure 6.10: *The trace-plot of emission probabilities to investigate the label switching between mixture components.*

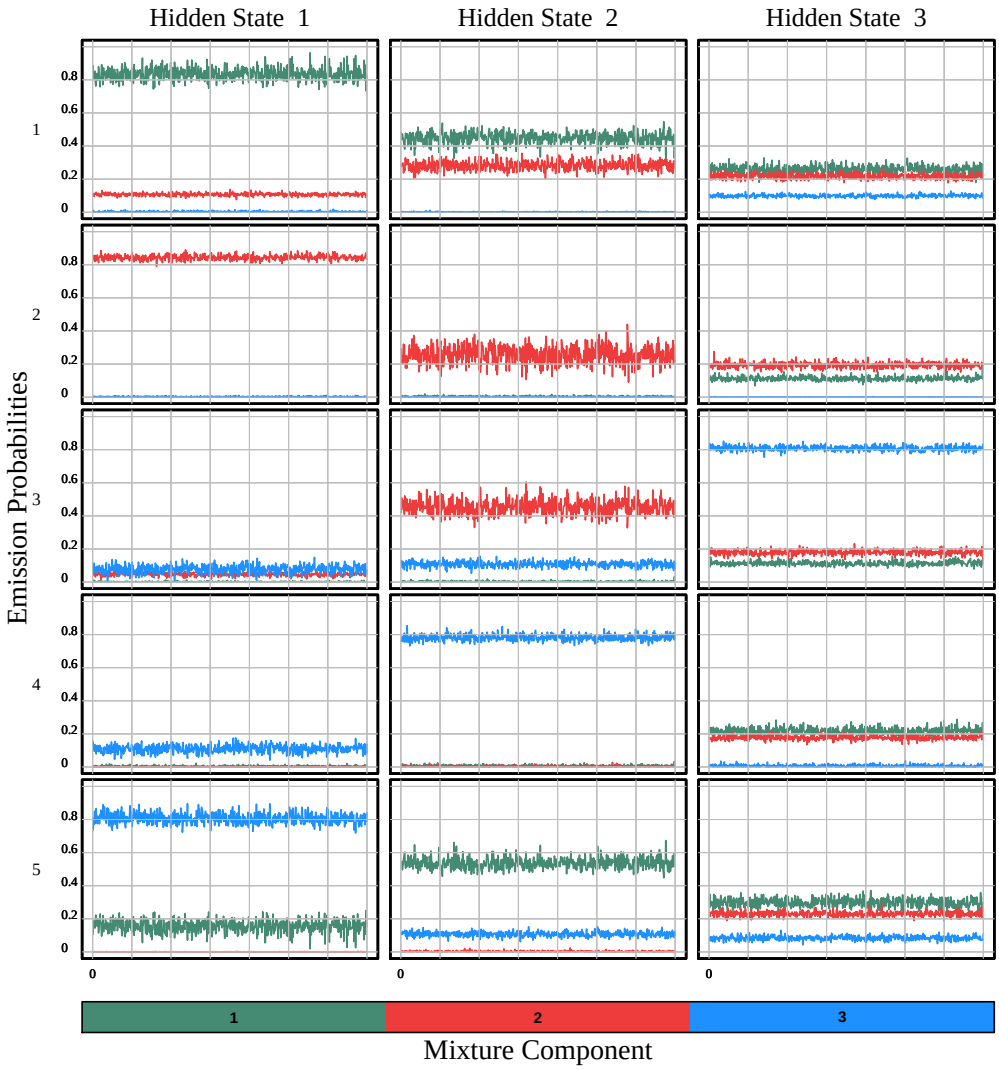
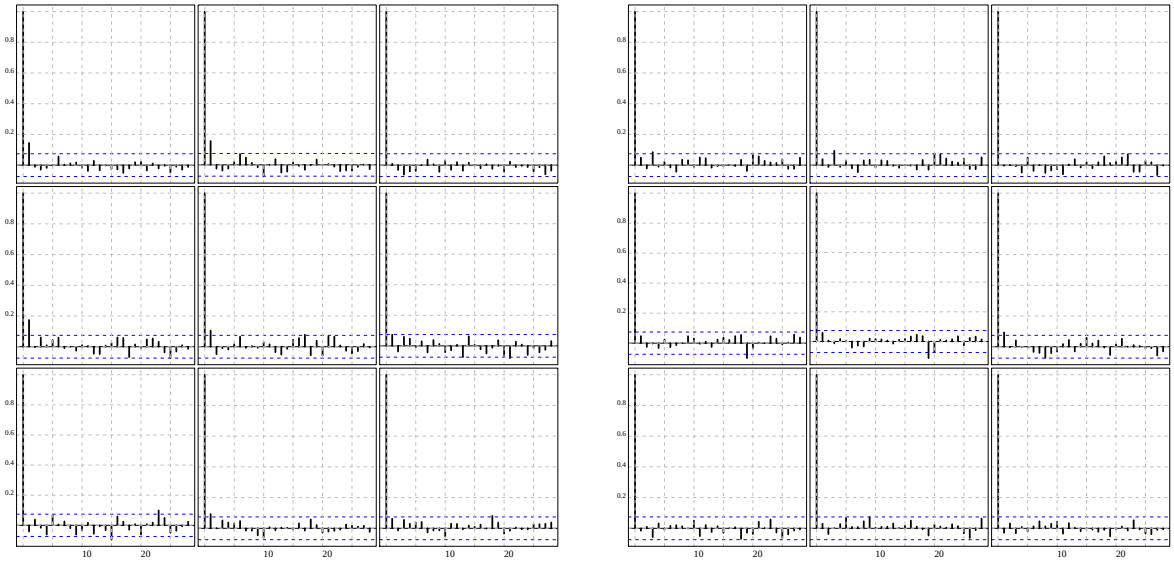


Figure 6.11: *The ACF plot of transition probabilities for the first component (left) and the second component (right).*



the number of components in the fitted model. Considering $K = 4$, we observed that the learned membership probabilities only assign sequences of observation to two mixture components of three. This can be considered as a way of finding that model has been misspecified.

6.8 Conclusions

MixHMMs are a flexible class of models useful for representing dependent heterogeneous phenomena. In this chapter we illustrated Bayesian inference for MixHMMs with a known number of regimes and components. We assumed the observed sequences are independent, conditional on the state variable, from a discrete distribution. We extended the MixHMM introduced by Ypma and Heskes (2002), which use the classical ML approach of inference in MixHMM, to a Bayesian analogue.

We considered a hierarchical model which allowed us to make vague a priori assumptions on the parameters of the model. The joint posterior distribution of all the parameters of the model was simulated by the Gibbs sampler. The updating of the parameters was illustrated. In particular, a stochastic forward-backward recursion was used to to improve the mixing of the chain compared to the direct Gibbs sampling algorithm. It also had

the advantage of providing efficient estimates of the posterior marginal distribution of the state variable itself. We illustrated how to choose an appropriate model for the data and obtain point estimates for its parameters on the basis of the MCMC output. Finally, we showed how the posterior predictive density of future observations can be simulated through the MCMC algorithm.

Chapter 7

Modelling Web Browsing: Bayesian MixHMM

In this chapter, we aim to model sequences of page requests within a session using a Bayesian MixHMM by a direct forward-backward Gibbs sampling method. The Bayesian mixture of hidden Markov model (BMixHMM) provides an approach to categorizing web pages into groups of web pages automatically. It also gives a way of grouping users into different clusters based on the web browsing pattern of visitors. We apply the MixHMM to the real clickstream data from a commercial website, and we obtain a rational page categorisation and user classification.

7.1 Introduction

A major concern in the context of web usage analysis has been to develop statistical models to describe the pattern of the sequence of web pages viewed whilst a user navigates through a website; the sequence of pages-viewed supplies information about the patterns of browsing behaviour of users through a website. Analysis of this kind of clickstream data, sometimes referred to as the page-view data set, enables the website maintainer to describe the common sequence in which web pages are viewed, or the category of related web pages. Website managers are usually interested in detecting common entrance/exit traces and page-visit routes that lead to visitors leaving the website, as well as some explanatory tools to improve the design of the website, ordering online, downloading, or any kind of action of interest. Page-view data sets may also help the website designer to personalise the website by determining different patterns of surfing through to the

website, usually by clustering visitors into different groups (Liu, 2007).

Page-view information can be extracted from the web server log-file. The key information on server log-files for this kind of data includes: the URL requested by the user, the IP address of the requesting computer and a time stamp delivered for the page request, mostly by clicking during navigation. This kind of data has a complicated structure as some visitors come back to the website for multiple visits. For this reason, the sequence of page-views is nested within a browsing session (Scott and Hann, 2007).

Although each visitor may have a browsing pattern, it is reasonable to assume that the aim of a visit may also affect browsing behaviour. For example, consider a typical visit to a website by a user who wishes to gather some information about a product he aims to buy. Next, he opens another browsing window (or tab) to gather more information from websites of other providers, whilst the Internet browser is still open for the previous one. He might then return to the original website to compare the information. This comparison may result in him adding an item to his basket through the website and visiting the conversion pages to order the product online, or going to the contact page to order by phone, or just leaving the website without any purchase.

Modelling browsing behaviour has been studied by researchers in different disciplines such as statistics, computer science, and marketing. Substantial amounts of early efforts in modelling user navigation patterns from web data were based on non-probabilistic techniques which try to find common navigation patterns, for example see Cooley et al. (1999b). One early work using the probabilistic approach to describe website navigation patterns was by Huberman et al. (1997). They applied the random walk approach to model the number of page-requests for a particular website. Markov models have been an essential approach for modelling the sequence of page-view observations (Jank and Shmueli, 2006). Sarukkai (2000) uses the first-order Markov chain as a tool for link prediction, so that each user is described by the first-order Markov model. This model helps to predict the most probable link a user may choose during their web browsing in the next session. A similar model is given later by Eirinaki, Vazirgiannis, and Kapogiannis (2005). Cadez et al. (2000a, 2003) used the mixtures of first order Markov chains as a model-based clustering approach for visitors to the website. Smyth (1997) used HMM as a probabilistic clustering approach for clustering individuals based on a sequence of observations. One approach is to model the browsing behaviour during each session, based on the sequence of web pages viewed regardless of previous sessions (Ypma and Heskes, 2002). Scott and Smyth (2003) applied the Markov Poisson process model to study the intensity of page requests within a web session. Kleinberg (2003) applied the point process to model patterns of browsing behaviour on a website. Montgomery et al. (2004)

introduced a hidden Markov model based on a dynamic multinomial probit regression to use both page-view data and also user-session data to model customer paths through an e-commerce website. Scott and Hann (2007) introduced a nested HMM, in order to describe the sequence of multiple sessions taken by the same user, to model the browsing behaviour in a specific session.

7.2 Modelling Browsing Behaviour

One advantage of using a stochastic model over other statistical analysis methods, such as the regression model or generalized linear model, is that it can be applied as an in-sessional analysis rather than post-sessional. This means that these models can be used to predict the ongoing browsing behaviour of users as they are surfing the web pages. The most desirable application is to distinguish between users with and without high potential to purchase online by tracing web page requests at each point of the session.

A typical website is made up of a large number of web pages. Hence, modelling the page-view sequences visited during a session induces a large number of parameters in the model. To avoid this problem the page-views may be modelled based on a sequence of *web page categories* instead of a sequence of web pages. However, manual categorizations can be cumbersome, as there is typically a large number of web pages on a typical E-commerce web site. In the case of modelling the browsing pattern by means of HMM, the large number of web pages increases the dimension of the emission matrices. This may not cause an over-fitting problem for the application over clickstream data sets, as there is always a sufficient amount of data available in this context. Nevertheless, it is desirable to assign each page to a specific category. The categorization is mainly based on the general purpose of the web pages and may differ from one website to another. An example of web categorization is given by Scott and Hann (2007) presented in Table 7.1. On the other hand, it is not usually an easy task to assign each of the web pages to a particular category. Note that a different choice of page categorisation may affect the resulting model. A solid web page categorization, in contrary to a probabilistic approach we use in this chapter, also might bring controversy, because there are usually some pages which can be assigned to more than one page category. Ypma and Heskes (2002) take the advantage of MixHMM as an approach for automatic (versus manual) categorization of web pages along with the inter-category transitions.

Users arrive at a website for different purposes. The aim of the visit may affect the pattern of browsing within the website. Hence, a web analyst is interested in assigning

Table 7.1: *Page categories in the page-view sequence data sets and their description (Scott and Hann, 2007).*

Category	Description
Home	Home page for the website, sometimes referred to as front-page of the website.
Product	Web pages devoted to presenting information about a single product.
Basket	Web pages regarding putting items into, or removing them from, the shopping basket/cart.
Information	Web pages which provide information such as: Frequently Asked Questions (FAQ), instructions on how to use the product, general information about the products, pages of legal disclaimers, etc.
Order	Web pages devoted to placing the order, relevant online forms, shipment, banking details, etc. These are sometimes referred to as shopping cart pages.
Shop Assist	Web pages devoted to displaying or comparing several products.
Site Map	Site Map for the online stores.
Contacts	Web pages containing general contact information including address, email, telephone, etc.
Exit	Exit/Leave web page is an artificial mark for the status in which a user leaves the website at the end of the session.

visit sessions to different types/categories. Several types of web session pattern may exist on a web site, and these types may differ from one website to another. Hence, introducing a model to distinguish the membership of a session to each of these patterns could be of great interest. This membership may be used as additional user data to describe web browsing behaviour. In this section we illustrate how the BMixHMM can be applied to label/cluster a session into different classes, as well as simultaneous web page categorisation.

7.2.1 Page-view Data Description

We used clickstream data collected during two weeks from an E-commerce website, where the focus was on the sequence of web pages viewed rather than the attributes of the web sessions. The website sells many different products which can be divided into four major categories: *electric shavers*, *dental care*, *health and well-being*, and *home applications*.

During the data collection period the site was not modified in terms of design or price. Some product offerings were changed but we assumed their effects to be negligible on general customer browsing behaviour. The website contains more than 120 distinct pages, but ultimate product pages have not been given in the log file, all being labelled by a single code. Therefore, there remain 40 web pages (or classes of web pages) distinguishable in the log-file, including the *exit* status. We also assigned each page a category label, based on the web page categories presented in Table 7.1. Single-page sessions that contained only one page were excluded from the data set - as mentioned earlier, single-page sessions are usually considered as either users who come to the website by mistake or automated computer programs which intrude on the website while scanning the web. The remaining 10091 sessions were generated by 8375 visitors and contained 126,348 page requests. We also added an *exit* page request at the end of each session sequence to distinguish between the return sessions. We split the data into training and testing data sets by choosing 4375 visitors at random for training and the rest were treated as a test data set. Training data was used to estimate the parameters of the model and test data sets were used to assess the predictive ability of the model for desired probabilities and labelling.

Table 7.2: *Sample of page sequence observation, webpages are coded from 1 to 40.*

User	Sequence
1	1 1 1 1 1 40 1 40 1 1 1 1 17 1 1 40
2	9 1 1 1 1 1 1 1 1 1 1 40
3	1 1 1 1 1 40
4	10 1 10 1 10 1 40
5	1 13 40 2 13 1 6 1 13 1 13 1 6 1 6 1 6 1 1 13 2 40
6	13 1 1 1 2 1 1 1 2 40
7	1 1 1 1 40
8	1 2 4 2 1 1 1 8 3 7 7 40
9	5 1 5 5 40
10	1 40 1 1 40 1 40 1 1 1 1 1 40 1 1 1 1 1 1 40 1 40

In addition to the sequence of page requests for each user, the time stamp supplying information about the time spent on page sequences, accurate to the nearest second was logged. Having completed the pre-processing tasks, including the user identification and sessionization, the log-file data files were converted into a set of sequences, where each sequence was represented as an ordered list of web page codes. We also used a different set of codes to represent the categories of web pages requested by the user. Table 7.2 shows a sample of such sequences for 10 users. The web-servers of the local website for a

twenty-four-hour period typically produce around 1000 such sequences.

Cadez et al. (2000b) introduced a plot to represent the sequence of web categories visited by users of a website. They call the it a *canvas plot* which consists of a sequence of coloured tiles, in which each tile represents a page request. Each row corresponds to a different user, and colours encode the category to which the requested page belongs. Figure 7.1 displays the sequence of web categories visited by 30 users. We used the same categories as given in the Table 7.1. The *Exit* category is an artificial page which shows the status of ending a session. Hence, the presence of the Exit block between sessions shows that the user returns to the website for repeat visits. The sessionization methods, used in the preprocessing, split the session so that it is ended when no page request occurs for more than 30 minutes. This may cause a pattern for users who leave the website open in the Internet browser.

A graphical representation of the empirical transition matrix between page categories (left) and corresponding entry page distribution (right) is depicted in Figure 7.2. A green spectrum represents the magnitude of the probabilities. The right panel shows the empirical probability of the entry page for each web page category, computed based on single-visits and the first session of repeat-visits. The entry page for most sessions is the *Product* page rather than the *Home* page. This is usually due to the links given through the search engine results when visitors search for products' names (Scott and Hann, 2007). A considerable percentage of visitors who begin the visit through the home page, either type the full IP address of the website, or come through the links provided by search engines.

Figure 7.2 (left) shows the empirical transition probabilities between web page categories. The last column of the table gives the probability of leaving the website from each category. The *contact* page category is followed by the highest frequency of website exit. This may be due to the fact that some visitors visit the contact page to place the order by phone; In this case, contact pages might serve as ordering pages. The *order* category has a high rate of exit (24%), showing that considerable percentages of users leave the website when an online order is submitted successfully. The graph also shows that about 1 out of 4 users leaves the website when faced with on error page. Error pages appear only in the ordering process. This mostly happens as ordering pages are hosted by a separate server from the rest of the web site, in order to offer greater security when providing information for a transaction on *Order* pages. So users who connect through weak internet services may therefore fail to complete the online transaction successfully.

The *product* category is the most frequent destination from other page categories on the

Figure 7.1: *The canvas-plot of the pages visited by 30 users based on the page categories introduced by Scott and Hann (2007)*

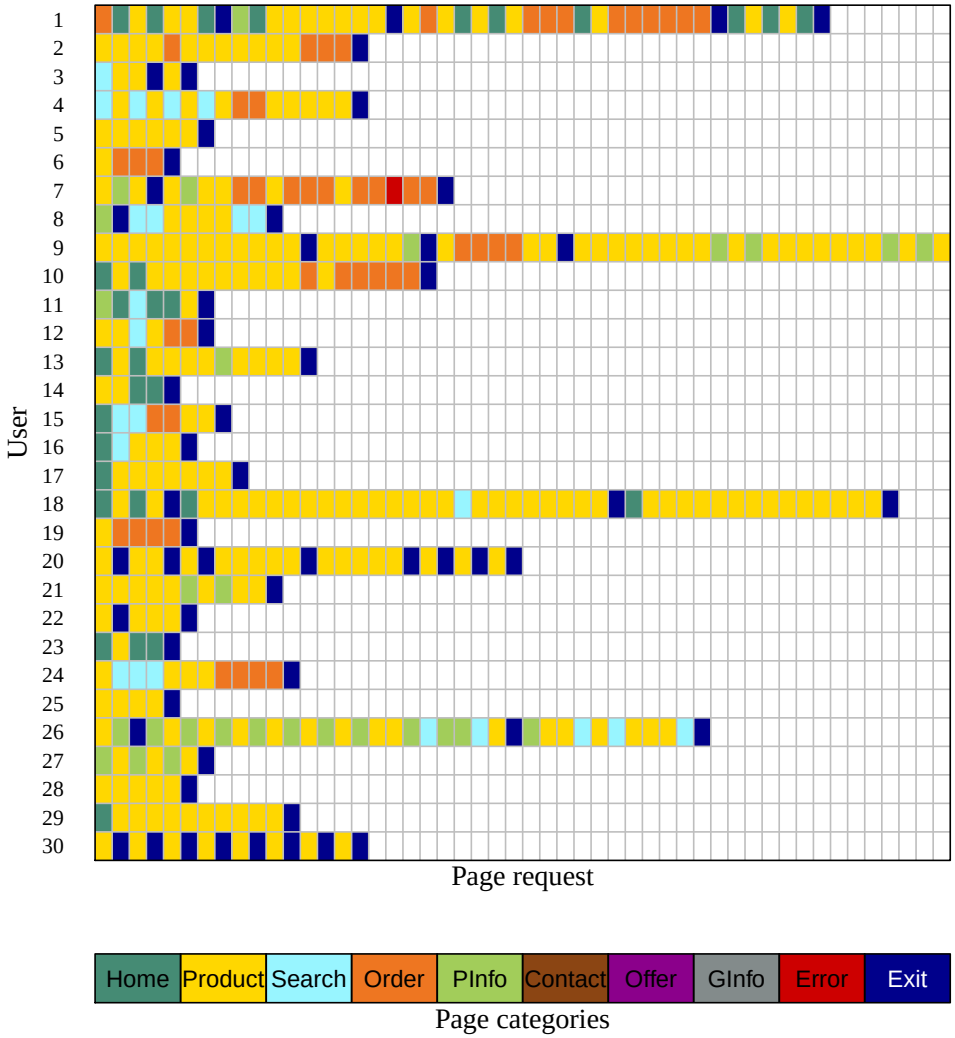


Figure 7.2: *Empirical probability transition matrix between page categories.*

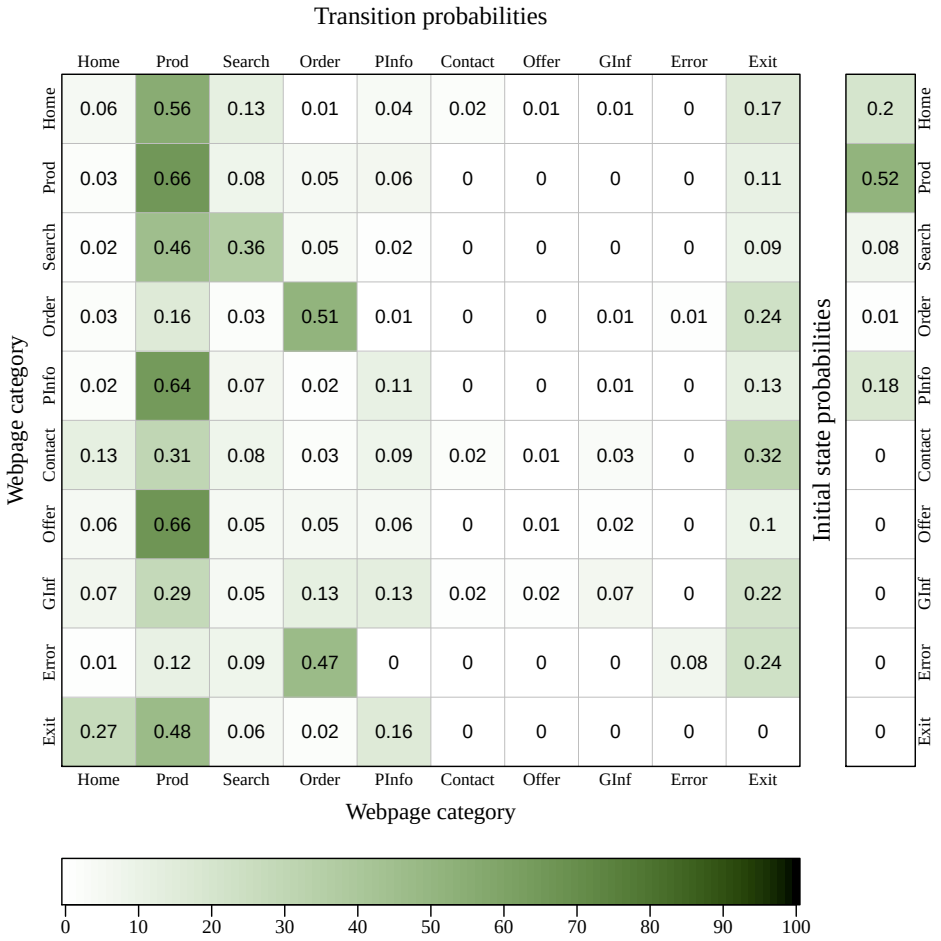
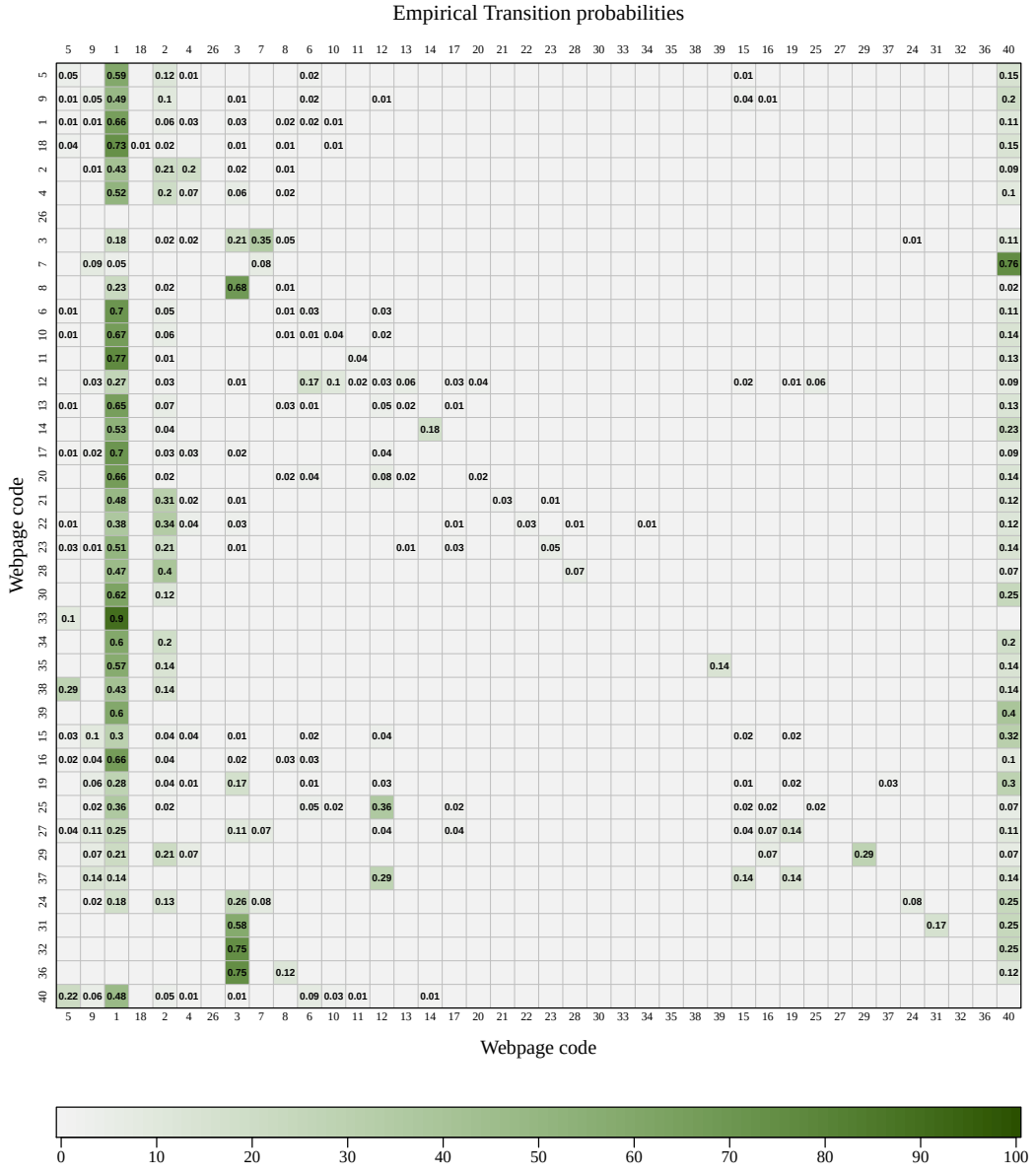


Figure 7.3: *Empirical probability transition matrix between web pages of the site*



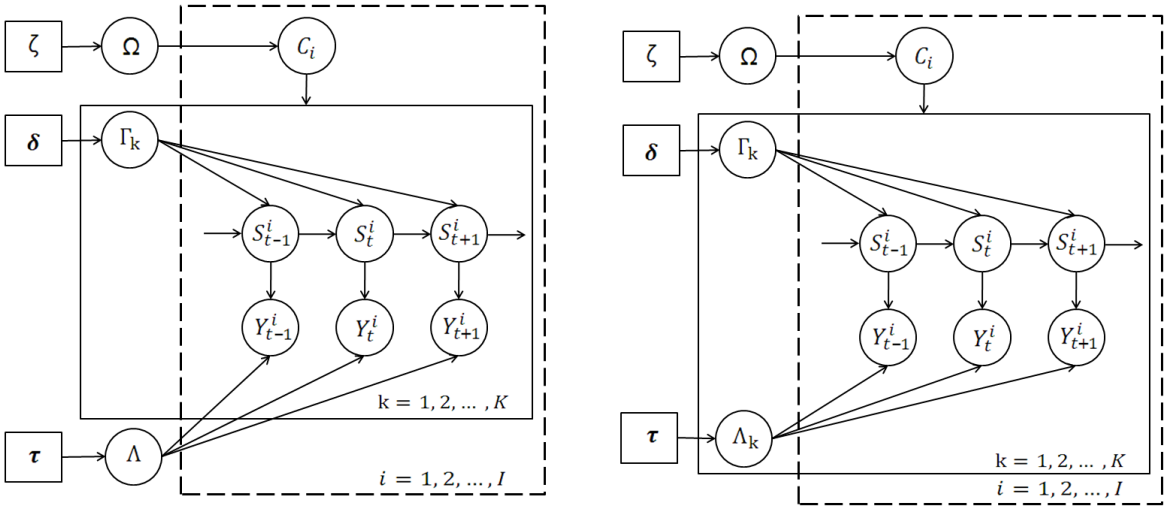
website, including a 66% probability of a self-transition. A long sequence of *product* page requests is a common pattern of shopping behaviour when customers compare the features of different products. In this example we have distinguished between web pages which provide general information rather than detailed information about the products. For example, newsletters, shipment information and refund policies are considered as *General information* (GInfo). The *product information* (PInfo) category includes pages providing information about features of the product/services which the E-Commerce website sells.

It is important to point out that transition probabilities in the last row, *Exit*, show the probability of entry page for visitors who return to the website for multiple visits. A comparison of these probabilities with entry page probabilities, given in the right-hand panel, helps to investigate the plausible different patterns of entering the website for multiple sessions versus single-visit sessions. For example, the percentages of entering the website through the home-page (27%) is larger for multiple sessions than first-visit sessions (20%). An explanation might be that visitors come back to the website by recalling the website in the address bar, using the bookmark feature of the Internet browsers, or typing the website address after the first visit.

Figure 7.3 displays the empirical transition probabilities between individual web pages of the website; the pages are labelled from 1 to 40. The last column of the table gives the probability of leaving the website from each page. We have also arranged the page labels in such a way that pages of the same category are next to each other. Since there is a large number of web pages, it is not an easy task to obtain a pattern from the empirical transition probabilities between web pages. However, one can find the high transition probabilities through the dark cells. For example, the *check-out* page has the highest frequency of leaving the website which supports the assumption that a considerable percentage of visitors leave the website after a successful online purchase. It also shows a high probability for exiting after visiting the *contact* page which may be due to the website's visitors ordering by telephone.

Some zero probabilities in Figure 7.3 are due to the topology of the website, in view of the fact that there exists no direct link between some web pages. In practice, one may find a small empirical transitions probability for structural disjoint pages. This is principally because of the proxy server caching ability of internet browsers by which a user can go back to previous pages by using the back button preserved in the internet browser. On the other hand, some automatic association between pages results in the completion of a transition. For example, it is common to invoke an automatic update of a page containing the users basket information so far when a user adds an item to the basket in the shopping cart pages.

Figure 7.4: *The DAG representation of the joint distribution of hierarchical Bayesian MixHMM for tied emission probability model (left) and untied model (right).*



7.2.2 Model Interpretation

In this section we apply the BMixHMM to describe the page-view pattern collected from a local E-commerce website. We illustrate the terminology used for the MixHMM in the web browsing equivalents. The joint probability distribution for two hierarchical Bayes models, the tied and untied models, is represented through DAGs in Figure 7.4. The DAG on the left assumes that the emission probabilities are the same for different components of the mixture model, usually known as the tied model in the context of HMM, but each component has a different transition probability matrix. Otherwise, the untied model assumes different emission probability matrices for different components, in addition to the different transition probability matrices. The joint distribution of all the variables conditional on fixed hyper-parameters is given by:

$$p(\Omega, \Gamma, \Lambda, \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{c} \,|\, \boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\zeta}) \propto p(\Omega \,|\, \boldsymbol{\zeta})\, p(\boldsymbol{c} \,|\, \Omega)\, p(\Gamma \,|\, \boldsymbol{\delta}, \boldsymbol{c})\, p(\Lambda \,|\, \boldsymbol{\tau}, \boldsymbol{c})\, p(\boldsymbol{s} \,|\, \boldsymbol{c}, \Gamma)\, p(\boldsymbol{y} \,|\, \boldsymbol{s}, \boldsymbol{c}, \Lambda). \quad (7.1)$$

We have earlier explained in chapter 6 how to use Gibbs sampling to take samples from posterior distribution of the parameters through three steps: (1) sampling from the membership state variable given the observation sequence and parameters; (2) sampling from the hidden states variables given the observation sequence and parameters; (3) sampling from the distribution of the parameters given hidden states, membership states, and observation sequence. This alternate sampling produces a sequence of N triplets of parameters, hidden state, and membership states $\{(\Theta, \boldsymbol{S}, \boldsymbol{C})_n, \quad n = 1, 2, \ldots, N\}$. Later in

this section, we explain how the parameters of the MixHMM can be interpreted in the web browsing behaviour context.

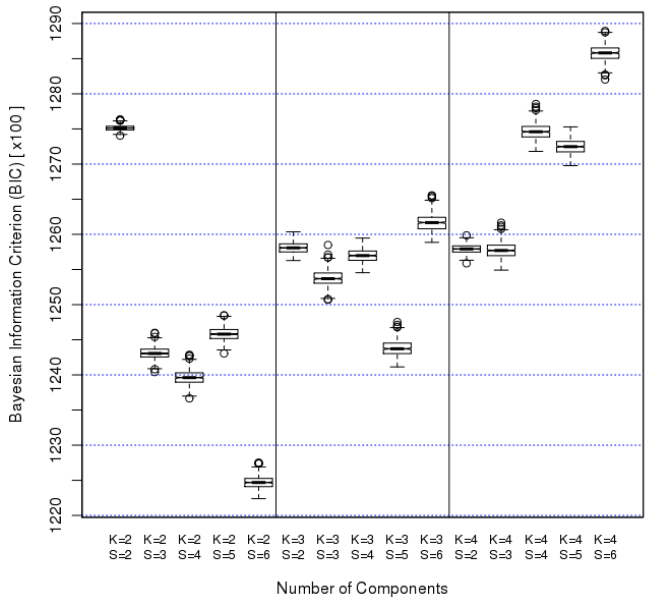
Let $\mathcal{S}_y = \{1, 2, \ldots, L\}$ denote the set of indices assigned to the web pages which make up the web site. We reserve the code L for the virtual exit/return status which is not an actual web page, but marks the end of a session, so that all sessions end with L. This is also used to represent returning to the website after an exit. As the same code is used for the status of the *exit* and *return* visit, no element of $\mathcal{S}_y$ is an absorbing state. The sequence of web pages viewed during a session by a user is denoted by $\boldsymbol{Y}_i = (Y_1^i, Y_2^i, \ldots, Y_{T_i}^i)$, where $Y_t^i \in S_y$ represents the t-th page request generated by the user i.

A visitor may visit a web page for various reasons, such as to obtain information about a product/service, to make an online order, to search for a product, to read the companys shipment policy, or merely to transition to another part of the website. Hence, a reasonable web page categorisation can be obtained using the intention of visitors who visit a web page. The application of MixHMM as a dynamic web page categorisation based on the intention of visiting a web page was first introduced by Ypma and Heskes (2002). They model each sequence of web browsing by a HMM of dimension S, where the hidden state variables are interpreted as the aim of visiting a web page during a web session. In other words, S_t^i is the aim of visiting the page of the code Y_t^i in the t-th page request by visitor i. The expression, *dynamic* web page categorisation means that the model helps to assign a value between 1 and S to S_i^t probabilistically.

It is also deemed that there are K distinct types of browsing behaviour which can be described by the components of the MixHMM. So, each session/user is assigned to one web browsing type. The hidden membership variable C_i serves as a label for the type of browsing behaviour associated with user i, as for each user we observe a different web browsing pattern, depending on the aim of the visit to the website in that session. It is assumed that visitors with different purposes will show their own specific pattern of viewing web pages. These labels are used to cluster users into different classes with respect to web browsing behaviour.

The transition matrix Γ_k gives the probability of transitions between web page categories for the user of type k. The emission matrix Λ comprises the probability of visiting different web pages when a user is in a specific web page category. In our application of MixHMM in which the hidden states produce a category over web pages, emission matrices are not expected to be too different for various sessions. For example, regardless of the type of visit, when the user is in the *product* category he/she is as likely to visit web pages that represent the products. Hence, we consider a MixHMM for which the emission matrix

Figure 7.5: *The boxplot of the BIC for different models fitted varies according to the number of clusters K and the number of hidden states S.*



is the same for all different user/session types. This restriction on the parameter space is usually referred to as *parameter tying* in the context of mixture models. In addition to the ease of interpretation for web page categories, parameter tying has the advantage of decreasing the risk of over-parametrisation in the model. Parameter tying is of more importance in the case of modelling the pattern of web browsing for websites containing a large number of web pages and small number of visitors. The DAG of the hierarchical Bayes model for MixHMM with tied emission matrices has been depicted in Figure 7.4. In this way, MixHMM shows differences between types of user through the contrast between transition matrices of components. In other words, classes of users are distinguished by patterns of movement between web categories.

The membership probability ω_k^i gives us the probability that a session/user i belongs to the type of session/user k. We summarise all membership probabilities into a $I \times K$ matrix denoted by Ω, where each row represents the user/session and each column corresponds to the mixture components of the model. The weight parameter ω_k shows the percentages of session/user which belongs to the type k. Equivalently, this is the membership probability for a typical session/user where there is no information about page-view sequence considered.

7.2.3 Model Selection

We first need to choose whether an untied MixHMM model is beneficial versus using a tied model. Fitting the untied model produces two emission matrices which necessarily does not gives the same web page categories. Hence, the result can not be of practical usefulness for web page categorization. Regardless of the application, using an untied model with S hidden states and K mixture components induces $(K-1) \times 40 \times S$ more parameters to the model in comparison with the untied model. This causes a considerable increase in BIC and consequently is not recommended based on a parsimonious principle for statistical modelling. Hence, for the rest of the chapter we fit and illustrate the results for the tied MixHMM.

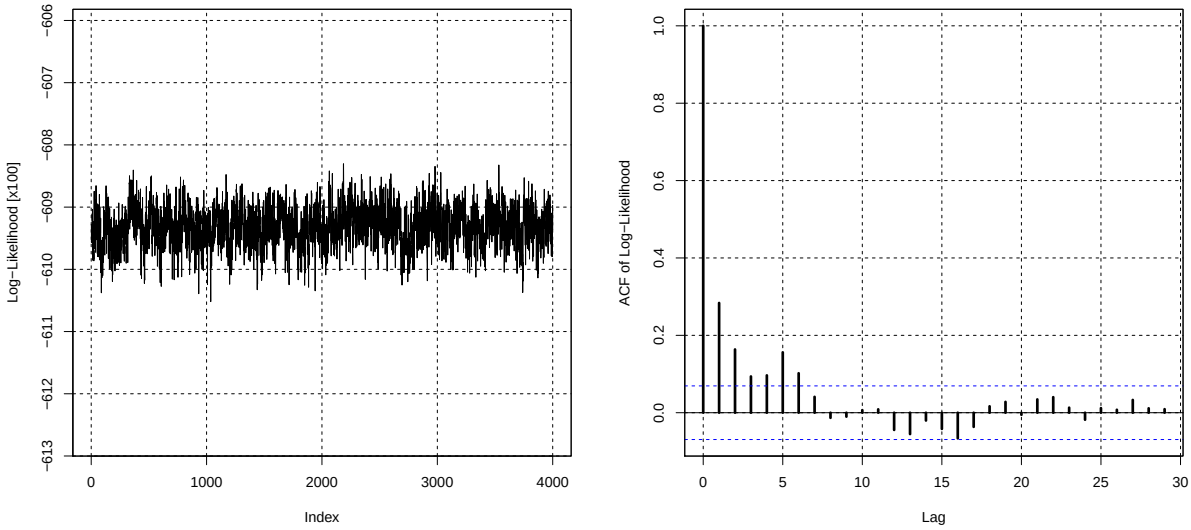
In order to determine the number of hidden states S and the number of components/clusters K, the Gibbs sampler is implemented for different combinations of $S \in \{2, 3, \ldots, 6\}$ and $K \in \{2, 3, 4\}$. The value of BIC for each model is recorded, and the model with the smallest BIC is chosen as a model which describes the browsing pattern the best compared to the rest. Figure 7.5 shows the box-plot of the posterior distribution of the BIC for fitted models. It can be seen that a MixHMM in a model with $(K, S) = (2, 6)$, provides the smallest BIC. There are 314 parameters, including two transition matrices Γ_k with 6×6 parameter, a common emission matrix Λ with 40×6 parameter, and two parameters in the mixture proportion vector $\boldsymbol{\omega}$. The difference between the BIC of this model and the rest is large enough to choose the simpler model and avoid the over-fitting problem. Therefore, the remainder of this chapter describes the model with $K = 2$ and $S = 6$, although we made similar exploration for the other choices.

7.2.4 Model Results

Figure 7.6 (left) displays the trace-plot for the Gibbs samples of the log-likelihood (in hundreds) of the model with $K = 2$ and $S = 6$. The caterpillar shape of the trace-plot shows that the samples of the conditional distributions of the parameters have reached the stationary status, or in other words, the sampler converged. Figure 7.6 (right) depicts the autocorrelation plot of the likelihood samples. This plot shows that the autocorrelation function declines rapidly. The autocorrelation will be much larger for the Gibbs samples without the use of forward-backward sampling and thinning.

Figure **??** shows the trace-plot of the posterior sample of transition probabilities for both mixture components, where different colours are used for different components. As in the trace-plot of the log-likelihood, the caterpillar shape of the graph shows that the samples

Figure 7.6: *The trace-plot of Log-Likelihood values of the function at each iteration (left) and the ACF plot of values of the log-likelihood.*



of the posterior distributions of the transition parameters have reached the stationary state. Using the same graph to represent mixture components helps to investigate the label-switching problem in mixture models; the paths reveal no evidence of a problem of label-switching of mixture components. This can be noticed in the graph by either swapping chains in the plot which cause changing colours or the appearance of multiple modes (Celeux et al., 2000). Figure **??** shows the MCMC sample paths of emission probabilities for all mixture components. To increase the visibility, the graph only shows the first eight emission probabilities, where different hidden states are represented by different colours. This graph also shows that the chain for emission probabilities has reached to the stationary state. It also shows that there is no label-switching problem at the level of hidden states.

The green spectrum is used to show the average of the estimated transition and emission probabilities. It should be noted that the hidden states need to be interpreted based on the emission probabilities, simply by looking at the conditional distribution of web pages, given that the chain is in a particular hidden state. Web pages with relatively high probability of being observed, when the chain is hidden state j, give us a clue to interpret the j-th web page category. Figure 7.9 represents the emission matrix of the fitted model, computed by taking an average over posterior samples of the emission probabilities. Using the tied model, the emission matrices for both mixture components are the same. In the

Figure 7.7: *The trace-plot of transition probabilities, as label-switching diagnostic.*

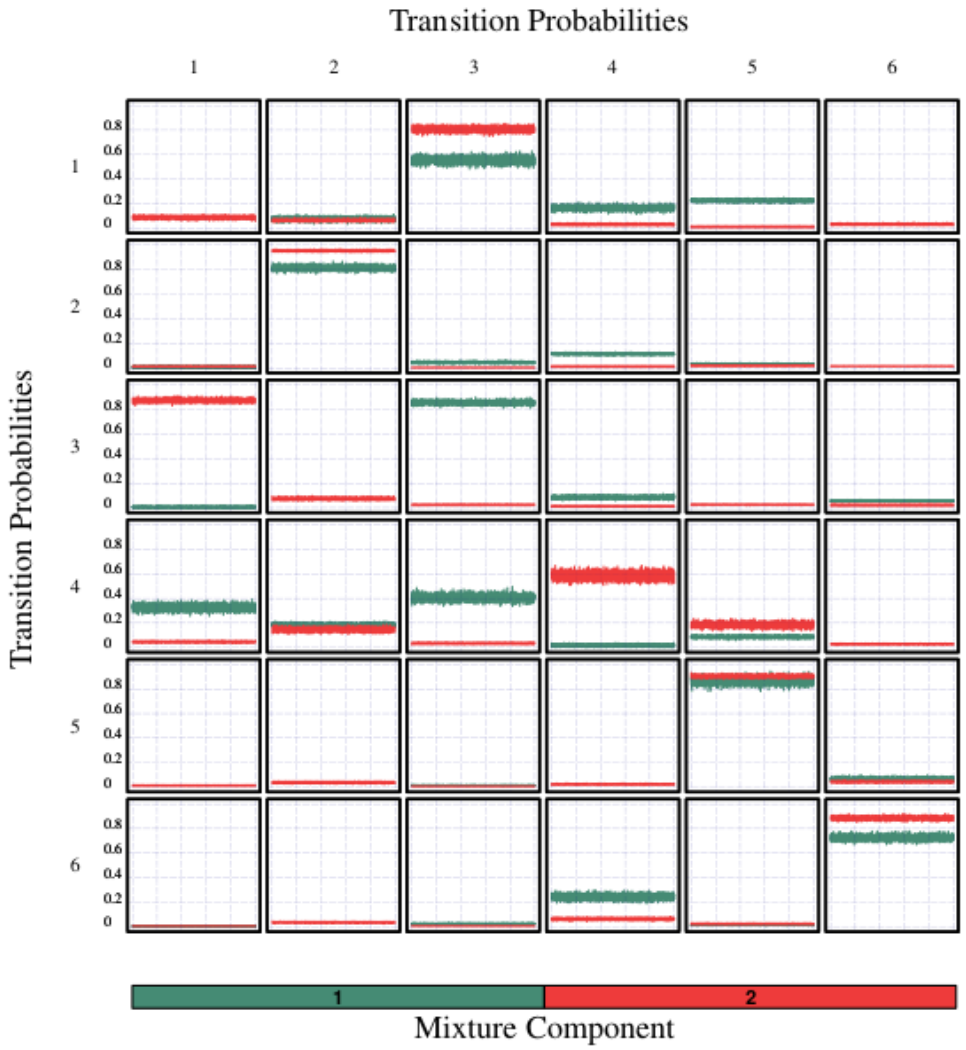


Figure 7.8: *Using the trace-plot of emission probabilities to investigate the label switching between hidden states. It only shows 8 emission states (web-page).*

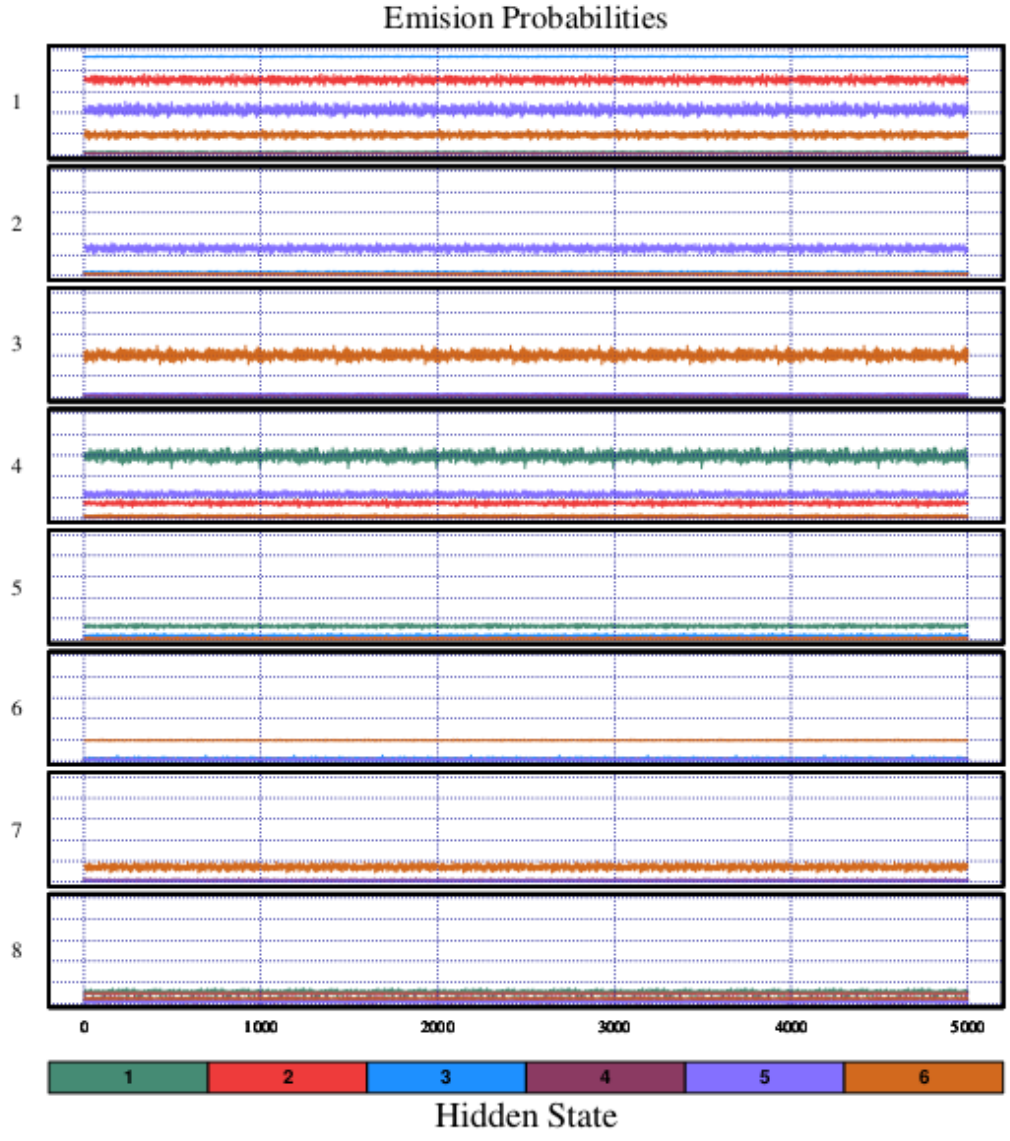


Figure 7.9: *Graphical representation of the emission matrix for tied MixHMM with $K = 2$ mixture components and $S = 6$ hidden states*

Emission probabilities Λ_1

	1	2	3	4	5	6
1	0.01	0.7	0.92	0	0.42	0.18
2	0.01	0.01	0.02	0.01	0.25	0.01
3	0	0	0	0	0.02	0.39
4	0	0	0	0	0.22	0.01
5	0.58	0.13	0	0	0	0
6	0.12	0	0.02	0	0	0
7	0	0	0	0	0	0.18
8	0	0	0	0	0	0.13
9	0.09	0.08	0	0.02	0	0.03
10	0.03	0	0.01	0	0	0
11	0.01	0	0	0	0	0
12	0.01	0.01	0	0.01	0	0
13	0	0	0	0	0	0
14	0.02	0	0	0	0	0
15	0	0.02	0	0	0.01	0
16	0	0	0	0.01	0	0
17	0	0.02	0	0	0	0
18	0	0.01	0	0	0	0
19	0.03	0	0	0	0	0.01
20	0	0	0	0	0	0
21	0.01	0	0	0	0.01	0
22	0.01	0	0	0	0.01	0
23	0.01	0	0	0	0.01	0
24	0	0	0	0	0	0.01
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0.01	0	0	0	0	0
30	0	0	0	0	0	0
31	0	0	0	0	0	0
32	0	0	0	0	0	0
33	0	0	0	0	0	0
34	0	0	0	0	0	0
35	0	0	0	0	0	0
36	0	0	0	0	0	0
37	0	0	0	0	0	0
38	0	0	0	0	0	0
39	0.01	0	0	0	0	0
40	0	0	0	0.91	0.04	0.01

same way we can show the the probabilities of observing each of the 40 pages of the website when the user is in hidden state $j = 1, 2, \ldots, 6$.

7.2.5 Interpreting the hidden states

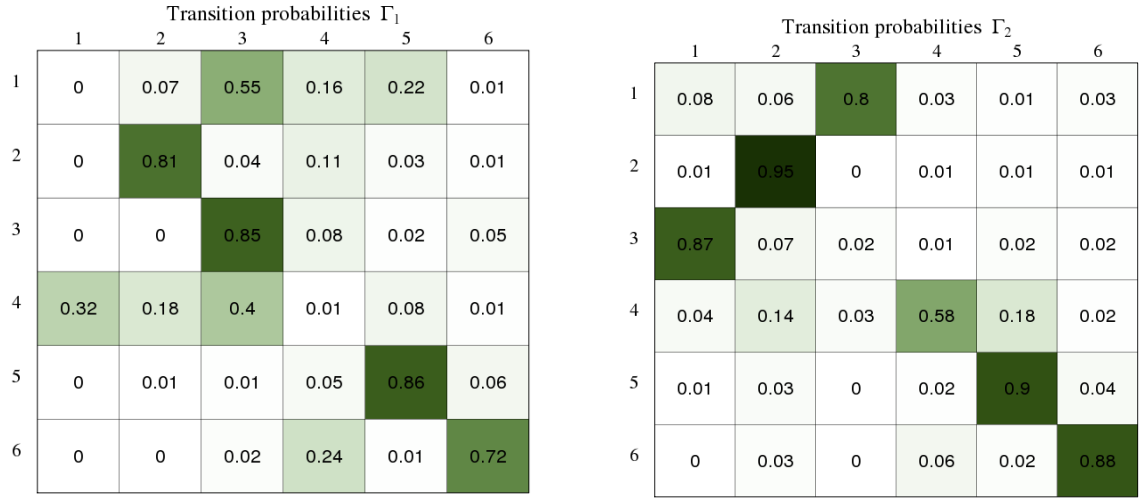
Let us now interpret these categories, referring back to Figure 7.9. The high probabilities of observation when the user is in the first state have been assigned to the front pages of the website such as homepage and guidances to choose between different types of main products. This implies that the first category can be considered as the *homepage category*. The second state is related to the pages which provide general information about the products for sale on an e-commerce website. State three assigns a very high probability, 0.92, for visiting the pages of items for sale, implying that the product pages are the main focus of this category. Having a very high probability on the virtual *exist/return* page shows that the fourth state represents the situation in which a user leaves the website. Fifth state represents search category, as it assigns the high probabilities to *search* and *sparessearch* web pages. The high probability of being in *product* pages for this category is because the *search* pages send the visitors to the *product* pages. The main focus for state six is on online shopping state so that it devotes large probabilities to the *Order/Basket*, *check-out-order* and related-items web pages. We summarise the these categories into the following table:

Hidden State	1	2	3	4	5	6
Description	Home	General Info	Product	Exit/Return	Search	Order

MixHMM produces categories of web-pages based on different kinds of action rather than solid classification. In practice, there is some support for this, as many page-clicks on websites produce artificial clicks. For example, by adding an item to the *basket* the website automatically shows the page of *related items* and also produces its corresponding records in the log file. There is usually a similar pattern for *search pages* in which users are guided to the product pages by searching by keyword. As a result, it is inevitable that there will be some product pages in the search category when the user is in the action of searching.

Figure 7.10 shows that two mixture components have a different pattern of transition probabilities between web page categories. The first row of the transition matrices reveals the transition to web categories from the homepage. It can be seen that users of the second browsing model tend to travel to product pages more than those in the first component.

Figure 7.10: *Graphical representation of transition matrices of tied MixHMM with $K = 2$ mixture components and $S = 6$ hidden states.*

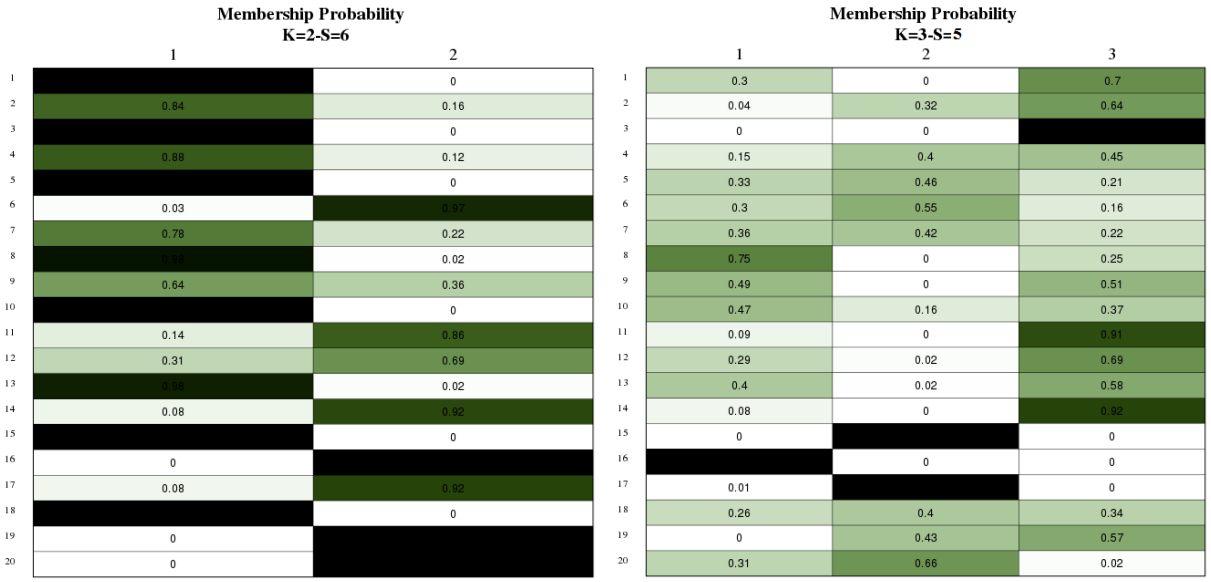


In contrast, users of the first pattern more likely to go to shopping-related web pages such as shopping cart and check-out pages, from home page rather than the second pattern (transition probability of $\gamma_1(1, 5) = 0.22$ versus $\gamma_2(15) = 0.01$).

Comparing the 4-th row of the transition matrices, corresponding to the return/exit category, provides some information about how users return to (or leave) the website for repeat visits in two browsing patterns. Figure 7.10 shows that the first browsing pattern has higher transition probabilities for all elements of the 4-th row, except for the 4-th column. This implies that there is a higher interest in returning to the website for repeat visit in the first browsing pattern, whilst a high probability of staying in state 4, $\gamma_2(4, 4) = 0.58$, implies that there are shoppers in the first browsing pattern who are willing to buy, as they add items into the basket, but may leave the website to check other online retailers, or gather information from other sources.

There is a fairly similar pattern of transition from search pages and shopping pages to other page categories for both mixture components. However, transition from shopping pages to exit/return is higher for the first component which can be explained by the fact that there are more online shoppers, who are more likely to leave and return to the website, in the first browsing pattern.

Figure 7.11: *Graphical representation of the membership probabilities for tied MixHMM model with $K = 2$ and $S = 6$ (left) and $K = 3$ and $S = 5$ (right).*



Clustering users by MixHMM

The MixHMM also provides a way of clustering users/session with the same browsing behaviour into different groups without predetermined web categories. This can be done by computing the membership probability for each user. Figure 7.11 shows the membership probabilities produced for 20 users/session who surf the website. In the left panel each row represents the average of the posterior sample of membership probabilities produced by tied MixHMM with $K = 2$ component and $S = 6$ hidden states. It can be seen that the model has been able to differentiate between user behaviour *reasonably*. This approach provides a soft clustering, when the model allows a unit to be assigned to several clusters partially, as well as hard clustering, when the unit should be assigned to only to one cluster. In the right panel we may compare the membership allocation for the equivalent tied MixHMM with $K = 3$ and $S = 5$, as the *best* of the models with three mixture components. It can be found that more clusters do not provide as good a separation for user groups, as there exist many users with nearly the same membership probability for different clusters. For example, observations 18 and 19 are well separated in the left panel, but the three-component model on right panel produces an unstable clustering decision between two or three components of the model.

Figure 7.12: *The posterior distribution for the probability of online purchase (left) leaving the website and return (right).*

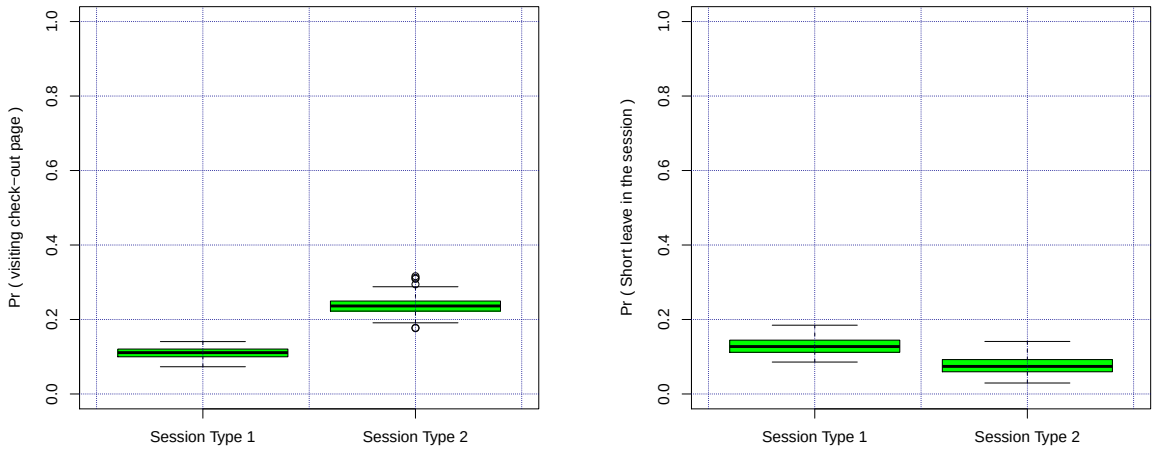
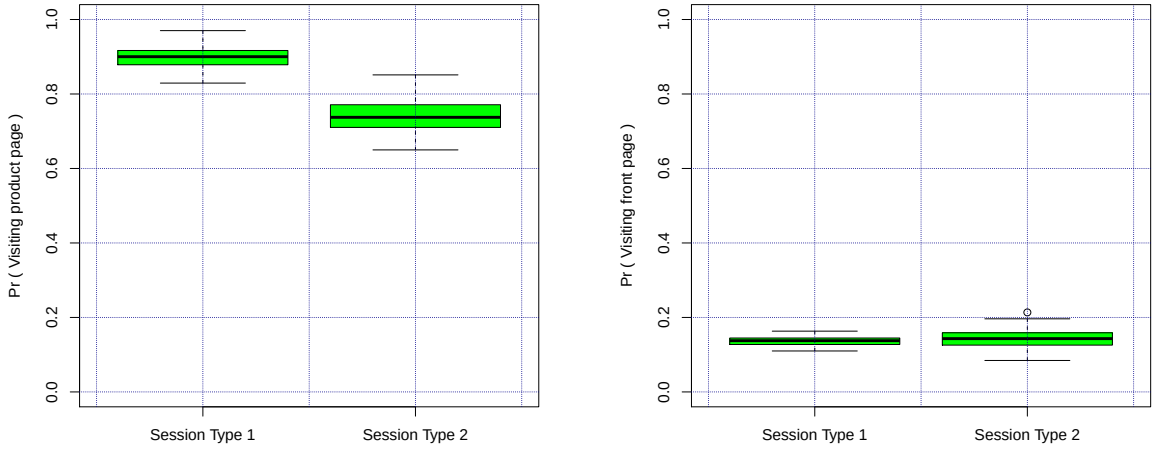


Figure 7.13: *The posterior distribution for visiting product pages (left) visiting front page (right).*



Investigating the session types

The estimate of mixture proportion parameters implies that 72% of all visitors have sessions of type 1, and the remaining are of type 2. Once we have the parameters of the MixHMM estimated, we are able to compute the probability of any pattern of page views. Calculating these probabilities also helps to make a better perception about different session types, represented by mixture components of MixHMM. As one of the most demanding events in the E-commerce context, we are interested in the probability of online shopping, and visiting the check-out-order page, based on the current pattern of the number of pages visited. The distribution of this probability is shown in Figure 7.12 (left) for each session type, corresponding to mixture components. It should be noted that this probability should be computed using the conditional probability of visiting the *check-out-complete* page given the number of pages viewed in the session. The odds ratio of online purchase in the second session type compared to the first session type is 2.50, so that a visitor of session type 2 is more than twice as likely to make an online purchase than a user of a session of type 1. The session types also differ according to the probability of returning to the website. Figure 7.12 (right) shows the probability that a user leaves a website and returns to continue the visit. Type 1 session clearly have a higher return probability than type 2, with an odds ratio of 1.8.

Figure 7.13 (left) shows a clear difference in the probability of observing product pages. The computed odds ratio, 3.05, states that users of type 1 are 3 times more likely to visit product pages than users of type 2. We are also interested in comparing the two session types in terms of coming to the website through the front page of the website. This probability is plotted in Figure 7.13 (right), where it is nearly the same for both types except the fact that the distribution of the posterior is more spread in session type 2. There might be several features in a web page traversal for which we can compute the posterior distribution of the probabilities for each session type. For example,the web owner might be interested to the found out what is the probability (percentage) of visiting the contact page, when a visitor has already added an item to the basket, or the probability of leaving the website without shopping after visiting the *shipment policy* page of the website.

7.3 Discussion and Future Work

We have used a Bayesian MixHMM approach for modelling the user behaviour on a website to produce dynamic categories of web pages and transitions between web page

categories simultaneously. The model also helps to assign user behaviour into different types/classes. We then display the behaviour of a random sample of users in each cluster along with the size of each cluster. The application of MixHMM provides reasonable web page categories for a local E-commerce pageview data set.

The proposed MixHMM can be extended into a multivariate case, where the sequence of time duration spent on each page helps with user behaviour clustering along with the sequence of web pages visited. This can be achieved by using duration models such as the log-normal or Weibull over the page visit time. Another extension which might improve the ability of the model to describe web browsing behaviour is characterized by considering a dependency structure among web pages within a given category in addition to the the transitions among web page categories.

A weighted analysis of the sequences might be applied by considering a sequence of weights jointly with the page view sequence. There are a number of methods which might be applied for weighting. An option is using the duration of visiting the page. Another possibility is to set the weight for the last page-view to be the mean time duration for the page taken across all sessions. It might be required to assign a normalized value of page duration instead of raw time. In some applications, the log of page time duration may help to reduce the noise in the data.

As mentioned earlier in chapter 6, reversible jump MCMC is an alternative to simultaneously estimate the posterior distribution of the number of components in the mixture model and the number of hidden states, and all other parameters of the model. A disadvantage of the using Gibbs sampler is that it is computationally expensive. The value of parameters at each step depends on the previous samples which suppress the use of multiple chains after convergence. Using the reversible jump makes it more computationally difficult and expensive. Hence, an improvement can be proposing a Bayesian methodology without heavy computational difficulties that enables researchers to make an inference about the dimension parameters, i.e. the number of hidden states and number of mixture components, while estimating the parameters of the model.

Chapter 8

Conclusion

This thesis originated from research questions proposed by experts of a local web management company which provides services for commercial websites selling products on the internet. Hence, the main purpose was to investigate statistical approaches on clickstream data, as the aggregate sequence of page visits executed by a particular user as the user navigates a website, can provide insight into the behaviour of visitors, specifically with respect to shopping, for e-commerce websites.

8.1 Clickstreams Data Preparation

We received two data sources: web log files and conversion data files. The log file contains information about every single click made by a user on a web browser while surfing the Internet, corresponding to an HTTP request sent to the server of the website. In addition to web logs, we received an operational data file for the websites, containing information on online purchases. We also produced a data format containing the sequence of page visits for every user as the user navigates a website. One limitation for the analysis of web page sequence is the presence of missing page references that are not recorded in the log-file. This is due to browser and proxy server caching. When a user returns to a page that has already been visited (downloaded) during the same session, the second access to that page will result in viewing the previously downloaded version of the page without sending a request to the server. This problem is usually solved by knowledge of the site structure. We performed a rule-based path completion procedure, but because of our limited knowledge of the website there is no guarantee that we were able to make a complete analysis.

The web log files do not consist of well-structured data and cannot be used directly for analytical purposes. Making clean clickstream data that can provide reliable information about web browsing behaviour requires a good understanding of the structure. For this reason, we explained the clickstream data structure and showed how to convert the raw data into data abstraction necessary for further analysis, referred to as a *data preprocessing* step, in the first chapter. For example, removing redundant records of log file data when a user requests a web page containing graphic and sound files, as the request results in several records/lines in the web log file that represent just one page request. We also needed to remove the records in the web log files made by bots, as those lines do not reflect human browsing behaviour.

For analytical purposes, we excluded bounce visits from the data, as they show the behaviour of visitors who have been guided to the website by mistake. A limitation on clickstream data is that there is no information available for the time spent on the last page visited. Considering the fact that time spent on a page provides critical information about the depth of the visit, statistical analysis of depth of visit suffers from the missing information. Another problem arises in clickstream data when the user actually leaves the session leaving the browser open. If the user then returns to the website through the open page in the browser, the clickstream data shows it as the same visit. We used the web sessionization rule of splitting the session into two when the time between two clicks is more than 30 minutes. User identification, which distinguishes between different users when a user visits a site more than once, has been implemented by the web management company.

Another limitation in using clickstream data is that log server data actually helps to distinguish between machines rather than users, except in the case of registered users of a website who logs into the website through a user ID and password. For example, if a user visits a website from a machine in the workplace, and later returns to the website from home, the user identification pre-processing fails to identify the user. Conversely, when a machine is used by several users, browsing the website by different people might be considered as a re-visiting of the website by the same user.

8.2 Exploratory Analysis of Clickstream Data

Throughout this thesis we have shown how clickstream data can provide an insight into the performance of a website and the behaviour of website visitors. This includes a brief review of web metrics and statistical reporting using clickstream data. Depending on the

goals of the analysis, this data could be transformed and aggregated at different levels of abstraction to provide metrics. For example, these metrics could be reported at the level of website or web session. We have illustrated that the metrics could provide valuable information to enable us to understand website usage and performance. In addition to the common application of clickstream for website traffic, this data could be used along with conversion information to help report some KPIs regarding the profitability of the website. As well as the metrics produced for the website, we reviewed the metrics that can be extracted from clickstreams, showing attributes of web sessions such as frequency, recency, and depth of a visit.

We also performed an exploratory analysis on depth-of-visit metric, by the number of pages visited and session time duration as metrics which indicate the depth of a visit. Most of the metrics in the clickstream data are less likely to follow the Normal distribution. Hence, the traditional statistical methods which assume the normality of the data need to be applied with caution. We showed that the number of pages visited can reasonably be approximated by Weibull distribution. We showed that the ML estimate of the parameters is not necessarily able to provide a reasonable estimate of the parameters, as it would make a very good fit for the high density part of the distribution and a poor fit for the rest. We showed that graphical estimates and PPCC provide an estimate of parameters so that the fitted model fairly explains the probabilistic behaviour of the variable at the dense part of the distribution, as well as the thin tail.

Despite the practical importance of the effect size for large sample circumstances, there has not been enough research on effect size measures. Effect size is mainly introduced and investigated for changes in location under an assumption of Normality for the underlying population. However, as mentioned earlier clickstream data are usually non-Normal. On the other hand, most non-parametric effect sizes measures compute dominancy measures, where each quantile of the distribution is larger than the equivalent quantile for the other one. This is not the case when the distribution depends on scale and shape and not just on a location parameter. For this reason, we introduced two novel alternatives ES for two-sample comparisons, one scale-free and one on the original scale measurement, and analysed some of their theoretical properties. We illustrated these measures by fitting a theoretical distribution, and also non-parametrically. We examined these ES for two-sample comparison studies under an assumption of Normality and Weibull distribution. The application of these effect sizes was represented for the session time duration in the customer behaviour analysis.

8.3 Conversion Analysis

Clickstream data typically contains information about the behaviour of visitors to a company's website. Investigating the behaviour of online buyers and non-buyers may provide a better understanding of the characteristics of visits with respect to shopping. The conversion analysis is of great importance for an e-commerce website manager. Consequently, there has been substantial interest in analysis of conversion using clickstream data. We used the logit model to describe the association between general clickstream information concerning visits and whether a visitor will engage in online-purchasing behaviour during his visit to the website. The ease of interpretation of the logit is an important advantage over other methods such as neural networks. Once the coefficient parameters are estimated, this model allows us to obtain a conditional probability estimate of purchase. The probability approximation can be used to rank customers in terms of their probability of purchase. This way, the web owner will be able to identify high potential visitors, in terms of conversion tendency, and generate leads for suitable targeting actions. In a web-focused marketing solution, the targeting action might help to keep the customer on the website rather than leaving to find another competitor.

Implementing an automatic stepwise model selection, we chose the best model, in terms of AIC, by choosing from the main effect model, as well as all possible interaction terms. In comparison with previous studies, our contribution has been to take into account interaction terms, as well as main effect general clickstream information. Our aim was to identify the most significant predictors of online purchasing to maximize the predictive power of our model in practice.

The results show that the predictive performance of the model increased significantly by considering the interaction terms in the model. The most important variables resulting from the selection techniques are the logarithm of the session time duration and whether the visitor is from the UK. The effect of more detailed regional information can be investigated in the case of availability of data. At the moment the model just looks at conversion regardless of the product. It would be interesting to model the conversion for different products/services that website offers.

There are some explanatory variables that are significant when performing a single logistic regression which were not entered into the final model. This is mainly due to the presence of multi-collinearity. The correlation matrix of the explanatory variables in the subset of chosen variables shows that there is not a large multi-collinearity which would be present when incorporating all of the explanatory variables. It should be noted that despite the predictive ability of the fitted logit model, there exists a severe discrepancy from the

assumptions of the model.

Comparing the result of the logit model with CART shows that the medium effect factors, session time duration and visiting from UK, appeared in both models. For the small effect model we would not necessarily have the same factors in the model. However, both models show very close predictive performance. It would be interesting to check the additional alternative approaches including neural networks, categorical principal component analysis (CATPCA) and further developments of CART models.

The model can be improved by considering more predictors in the model, for example any knowledge about the web pages viewed during the session. We may be interested in distinguishing between the number of content pages versus the number of transition pages visited during a session by a visitor and determining whether it affects the probability of conversion. Time spent on reading the shipping policy might be associated with the probability of conversion. Conversely, we may find some pages that help to identify visitors who are on the website to gather information, for example visitors who download white papers not specifically related to the service or products that the website offers. Using registered users who login to the website, depending on the requested information in the registration forms, may increase the predictive performance of the model. This includes attributes such as gender, age, occupation, and educational level.

8.4 Analysis of Sequences of Pages Visited

The ordering of pageviews visited by users will also provide information about their browsing behaviour. A web manager might be interested in analysing the clickstream path taken by users during their session on the website. Furthermore, clustering of users can be implemented based on methods which consider the ordering of pageviews. Since the early stages of web usage analysis, mixture models have been proposed as model-based clustering approaches for clustering users. The general idea behind mixture models, such as a mixture of Markov models, or mixture of hidden Markov models, assumes that K user clusters exist (user behaviour in this context) within the data. Each user session is assumed to follow a probability model of the observed (and hidden variables for MixHMM). In this approach, a user cluster is chosen with some probability. Then, the user session is generated from a Markov model with parameters specific to that user cluster.

A mixture of Markov models can not only probabilistically cluster user sessions based on similarities in navigation behaviour, but can also capture characteristics of each user cluster, as it enables us to compute the probability of visiting a page or any sequence

of pages. In our application we computed the probability of contact page, check-out or product pages for different clusters. Model-based clustering techniques can be applied to other session attributes, such the time spent on each page. For example, Mair and Hudec (2008) introduced the mixture of proportional hazard models to group navigational behaviour with respect to sequence of time spent on webpages.

We reviewed the theoretical background of hidden Markov models and mixture of hidden Markov models. We extended the EM algorithm for the MixHMM with observations from the Normal, Poisson, Exponential, and Binomial distributions. The classical EM algorithm for MixHMM does not provide an estimate of the variance of the parameters. On the other hand, different plugin values of the EM algorithm may result in a different ML estimate of the parameters. So we used the Bayesian approach with vague prior distribution over parameters of MixHMM to find an estimate of the parameters, as well as their precision.

We provided the theoretical background to implement the mixture of Hidden Markov models (MixHMM) in the Bayesian framework. We applied this model for modelling internet browsing behaviour on an e-commerce website. As the aim was to model web-page traversal we focused on the output of hidden Markov Models with discrete nominal distributions. We assumed that the number of components and the number of regimes for the hidden Markov models were known.

The main difficulty of using Gibbs sampling for the MixHMM, as a high dimensional model, was the high level of autocorrelation for samples, and consequently slow mixing. As an early solution in the mixture model we used a stochastic forward-backward recursion to improve the mixing of the chain compared to the direct Gibbs sampling algorithm. Additionally, we used thinning to provide an independent sample from the marginal posterior distribution of the model parameters. The performance of the model was assessed over an artificial navigation pattern. It should be noted that diagnostic checks are not very well developed in the HMM and consequently for MixHMM. Therefore further improvement could be made in terms of providing diagnostic checks for the MixHMM.

Further development could be achieved by providing a methodology that would allow us to estimate the number of mixture components and hidden states while estimating the parameters of the model. The reversible jump MCMC technique has already been proposed to simultaneously estimate the posterior distribution of the number of regimes in the mixture model and all other parameters of the model. However, we found it very complicated to implement it for MixHMM due to computation of the Jacobian matrix. Hence, it would be desirable to find a simpler methodology for computing the number

of mixture components for high dimensional models such as MixHMM. Our approach at the moment looks at the distribution of BIC for running the model for several different settings. This approach is computationally expensive, as the Gibbs sampler needs a considerable number of samples to provide an adequate independent sample.

Furthermore, the large sample size nature of the clickstream data induces some difficulties in using standard statistical methods such as the two-sample t-test, and goodness-of-fit test, as the test is significant even for very small effects. The classic goodness-of-fit tests are often significant, although, the graphical tools reveal a small departure of the observed and the theoretical distribution. This suggests that we should equip our analysis based on the adequate graphical tools, as well as effect size measures to provide the analyst with an alternative to assess practical significance along with statistical significance.

The proposed MixHMM assumes that observing each page only depends on the hidden status of the user at time t and its type of session. It would be beneficial to consider the effect of the previous page on the next page, as well as the hidden state. We showed that MixHMM can serve as a model-based approach to clustering sessions. However, this model only considers the transition between web-pages. An improvement could be obtained by using a methodology which takes into account other general features of a web session (or users) such as session time duration, number of pages visited, traffic source, etc. to determine the groups of session/users that are close to each other based on a measure of distance or similarity. Other types of analysis that can be performed on sequential patterns include trend analysis, change point detection, or similarity analysis.

Another idea would be to develop a multivariate Mixture of hidden Markov model, which would consider the time spent on each page, as well as page traversal. When we attempt to use two variables in the model, an alternative is to consider the effect of time on the pages on the MixHMM by developing a weighted version of MixHMM. Montgomery et al. (2004) introduce the dynamic multinomial probit model to model online browsing behaviour, engaging pageview sequence in cooperation with demographic information as well as content measures.

Finally, we wish to stress that advances in data pre-processing, analyzing, and modelling, to the clickstream data is very demanding at the moment. The successful application of these models has been found in different web personalization, customer relationship, PPC optimisation, etc. Despite this progress, developing robust and effective methodology is still in demand.

Bibliography

R.B. Abernethy, J.E. Breneman, C.H. Medlin, and G.L. Reinman. Weibull analysis handbook, 1983.

A. Abraham. Business intelligence from web usage mining. *Journal of Information and Knowledge Management*, 2(4):375–390, 2003.

A. Agresti. *Categorical Data Analysis*. New York: John Wiley & Sons, 1990.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716723, 1974.

J.H. Albert and S. Chib. Bayes inference via gibbs sampling of autoregressive time series subject to markov mean and variance shifts. *Journal of Business and Economic Statistics*, 11:1–15, 1993.

J. Algina, H.J. Keselman, and R.D. Penfield. An alternative to cohen's standardized mean difference effect size: A robust parameter and confidence interval in the two independent groups case. *Psychological Methods*, 10:317–328, 2005.

J. Andersen, R.S. Larsen, A. Giversen, T.B. Pedersen, A.H. Jensen, and J. Skyt. Analyzing clickstreams using subsessions. Technical report tr-00-5001, Department of Computer Science, Aalborg University, 2000.

C. L. Anderson and J. C. Berry. On a simple measure of dominance. *Journal of Statistical Planning and Inference*, 139:1098–1108, 2009.

P.C. Austin. A comparison of regression trees, logistic regression, generalized additive models, and multivariate adaptive regression splines for predicting ami mortality. *Statistics in Medicine*, 26:2937–57, 2007.

B. Baesens, G. Verstraeten, D. Van Den Poel, M. Egmont-Petersen, P. Van Kenhove, and J. Vanthienen. Bayesian network classifiers for identifying the slope of the customer

- lifecycle of long-life customers. *European Jounal of Operational Research*, 156(2): 508–523, 2004.

D. L. Banks and Y. H. Said. Data mining in electronic commerce. *Statistical Science*, 21 (2):234–46, 2006.

P. Bapna, R.and Goes, R. Gopal, and J. R. Marsden. Moving from data-constrained to data-enabled research: Experiences and challenges in collecting, validating and analyzing large-scale e-commerce data. *European Journal of Operational Research*, 21(2): 116–130, 2006.

R. E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing: Probability Models*. New York: Holt, Reinhart and Winston, 1975.

H.H. Bauer, M. Grether, and M. Leach. Building customer relations over the internet. *Industrial Marketing Management*, 31(2):155–63, 2002.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

B. Berendt, Mobasher B., M. Nakagawa, and M. Spiliopoulou. The impact of site structure and user environment on session reconstruction in web usage analysis. In *KDD02 WebKDD Workshop*, 2002.

P. Berkhin, J.D. Beche, and D.J. Randall. Interactive path analysis of web site traffic. In D. Lee, M. Schkolnic, F. Provost, and S. Ramakrishnan, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD2001*, pages 414–419, San Francisco, CA, USA, August 2001. MIT Press.

G. E. P. Box and D. R. Cox. An analysis of transformations (with discussion). *Journal of the Royal Statistical Society B*, 26:211–252, 1964.

L. Breiman, J. Friedman, and C. Stone. *Classification and regression trees*. Chapman and Hall, 1984.

T.S. Breusch and A.R. Pagan. Simple test for heteroscedasticity and random coefficient variation. *Econometrica*, 47(5):1287–1294, 1979.

G. Brys, M. Hubert, and Struyf A. Robust measure of tail weight. *Computational Statistics & Data Analysis*, 50(3):733–759, 2006.

A. Buchner and M. D. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. In *ACM SIGMOD International Conference on Management of Data (SIGMOD99)*, pages 54–61, 1999.

R. E. Bucklin and C. Sismeiro. A model of web site browsing behavior estimated on clickstream data. *Journal of Marketing Research*, 40:249–267, 2003.

R. E. Bucklin and C. Sismeiro. Click here for internet insight: Advances in clickstream data analysis in marketing. *Journal of Interactive Marketing*, 23(1):23–48, 2004.

R.E. Bucklin, J.M. Lattin, A. Ansari, S. Gupta, D. Bell, E. Coupey, J.D.C. Little, C. Mela, A. Montgomery, and J. Steckel. Choice and the internet: From clickstream to research stream. *Marketing Letters*, 13(3):245–258, 2002.

W. Bucklin and D. Van den Poel. A model of web site browsing behaviour estimated in clickstream data. *Journal of Marketing Research*, 40(3):249–267, 2003.

I. Cadez, D. Heckerman, C. Meek, P. Smyth, and White S. Visualization of navigation patterns on a web site using model-based clustering. Technical report, University of California, Irvine, March 2000b.

I. V. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals and objects, 2000a.

I. V. Cadez, D. Heckerman, Meek, P. C., Smyth, and S. White. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining Knowledge Discovery*, 4(4):399–424, 2003.

G. Casella. Leverage and regression through the origin. *American Statistician*, 37(2): 147–152, 1983.

R. Castellano and L. Scaccia. Bayesian inference for hidden markov model. Working Papers 43–2007, Macerata University, Department of Finance and Economic Sciences, Oct 2007. URL `http://ideas.repec.org/p/mcr/wpdief/wpaper00043.html`.

L. Catledge and J. Pitkow. Characterizing browsing strategies in the world wide web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, 1995.

G. Celeux, M. Hurn, and C.P. Robert. Computational and and inferential difficulties with mixture posterior distributions. *Journal of Amerian Statistical Association*, 95(451): 957–970, 2000.

S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann, 2003.

S. Chakrabarti, B. Dom, P. Raghava, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *In Proceedings of the Seventh International World Wide Web Conference*, pages 65–74, 1998.

J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth & Brooks/Cole, 1983.

J.M. Chambers and T.J. Hastie. *Statistical Models in S*. Chapman & Hall, 1993.

P. Chatterjee, D. L. Hoffman, and T. P. Novak. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22(4):520–541, 2003.

S. Chib. Calculating posterior distributions and model estimates in markov mixture models. *Journal of Econometrics*, 75:79–97, 1996.

G.A. Churchill. Stochastic models for heterogeneous dna sequences. *Bulletin of mathematical biology*, 59(1):79–94, 1989.

W. S. Cleveland. *The elements of graphing data*. Murray Hill, N.J: AT&T Bell Laboratories, 1994.

N. Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114:494–509, 1993.

R. Coe. Its the effect size, stupid. what effect size is and why it is important. In *Proceedings of the the Annual Conference of the British Educational Research Association*. University of Exeter, England, September 2002.

J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Academic Press, revised edition edition, 1977.

J Cohen. A power primer. *Psychological Bulletin*, 112:112–159, 1992.

D. Collett. *Modelling Binary Data*. London: Chapman and Hall, 1991.

Douglas Comer. *Internet working with TCP/IP: Principle, Protocols,a nd Architectures*. Prentice Hall, 4 edition, 2000.

R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999a.

R. Cooley, P.N. Tan, and J. Srivastava. Discovery of interesting usage patterns from web data. In *Proceedings of the 1999 KDD Workshop on Web Mining*, 1999b.

R. Cooley, P. N. Tan, and J Srivastava. Discovery of interesting usage patterns from web data. In *In Proceedings of web usage analysis and user profiling*, pages 163–182, 2000.

D.R. Cox and E.J. Snell. *The Analysis of Binary Data*. London: Chapman and Hall, second edition, 1989.

P. J. Danaher, G. W. Mullarkey, and S. Essegaier. Factors affecting website visit duration: A cross-domain analysis. *Journal of Marketing Research*, 43(2):182–194, Sepember 2006.

R. B. Darlington. Comparing two groups by simple graphs. *Psychological Bulletin*, 79(2): 110–116, 1973.

A.C. Davison and D. Hinkley. *Bootstrap Methods and their Application*. Cambridge, 8th edition edition, 2006.

E. Demers and B. Lev. A rude awakening: Internet shakeout in 2000. *Review of Accounting Studies*, 6:331–359, 2001.

J. Descôteaux. Statistical power: An historical introduction. *Tutorials in Quantitative Methods for Psychology*, 3(2):28–34, 2007.

J. G. Dias, J. K. Vermunt, and S. Ramos. *Mixture Hidden Markov Models in Finance Research*, pages 451–+. 2010.

K. A. Doksum. Some graphical methods in statistics. a review and some extensions. *Statistica Neerlandica*, 31:53–68, 1977.

K. A. Doksum and A. Samarov. Nonparametric estimation of global functionals and a measure of the explanatory power of covariates in regression. *Annals of Statistics*, 23: 1443–1473, 1995.

K. A. Doksum and G. L. Sievers. plotting with confidence - graphical comparisons of 2 populations. *Biometrika*, 63:421–434, 1976.

N. R. Draper and H. Smith. *Applied Regression Analysis, 3rd Edition*. John Wiley & Sons, 1998.

X. Dreze and F. X. Hussherr. Internet advertising: Is anybody watching? *Journal of Interactive Marketing*, 17(4):8–23, 2003.

M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, 2003.

M. Engelhardt. On simple estimation of the parameters of the weibull or extreme-value distribution. *Technometrics*, 17(3):369–374, 1975.

W. W. Esty and J. D. Banfield. The box-percentile plot. Technical Report 63, Department of Mathematical Sciences, Montana State University, May 1992.

B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman & Hall, 1981.

P. S. Fader, B. G. S. Hardie, and K.L. Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4):415–430, 2005.

J. J. Filliben. The probability plot correlation coefficient test for normality. *Technometrics*, pages 111–117, February 1975.

T. R. Fleming, O'Fallon, J. R., O' Brien, P. C., and D. P. Harrington. Modified kolmogorov-smirnov test procedures with applications to arbitrarily right-censored data. *Biometrics*, 36:607–625, 1980.

S. Fruhwirth-Schnatter. Markov chain monte carlo estimation of classical and dynamic switching and mixture models. *Journal of the American Statistical Association*, 96 (453):194–209, 2001.

C. Fuh and I. Hu. Estimation in hidden markov models via efficient importance sampling. *Bernoulli*, 13:492–513, 2007.

Casella G. and E.I. George. Explaining the gibbs sampler. *The American Statistician*, 46 (3):167–174, 1992.

A. Gelman and J. Hill. *Data Analysis Using Regression Multilevel/Hierarchical Models*. Cambridge Press, 2006.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Learning*, 6(6):721–741, 1984.

J.D. Gibbons and S. Chakraborti. *Nonparametric statistical inference*. CRC Press, 4th edition, 2003.

G. Gigerenzer. *A Handbook for Data Analysis in the Behavioral Sciences: Methodological Issues*, chapter The Superego, the Ego, and the Id in Statistical Reasoning, pages 311–339. Hillsdale, NJ: LEA, 1993.

W.G. Gilchrist. *Statistical Modelling with Quantile Functions*. Chapman and Hall, 2000.

N. Glady, B. Baesens, and C. Croux. Modeling churn using customer lifetime value. *European Journal of Operational Research*, 197(1):402 – 411, 2009.

G.V. Glass, B. McGaw, and M.L. Smith. *Meta-analysis in social research*. Beverly Hills, CA: SAGE Publications, 1981.

P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

R. J. Grissom. Probability of the superior outcome of one treatment over another. *Journal of Applied Psychology*, 79(2):314–316, 1994.

R. J. Grissom and J. J. Kim. *Effect Size for Researches: A Broad Practical Approach*. Psychology Press, Taylor and Francis Group, 2005.

W. A. Hanson. *Principles of Internet Marketing*. Cincinnati: South-Western College Publishers, 2000.

D. Harte. Mathematical background notes for package hiddenmarkov. Technical report, 2010. http://homepages.paradise.net.nz/david.harte/SSLib/Manuals/notes.pdf.

L. V. Hedges and L. Friedman. Gender differences in variability in intellectual abilities: A reanalysis of feingold's results. *Review of Educational Research*, 63(1):94–105, 1993.

L.V. Hedges. Distribution theory for glass's estimator of effect size and related estimators. *Journal of Educational Statistics*, 6(2):107–128, 1981.

M. Helmy, M. Norzali, H. Fahri, and M. Farhan. Data pre-processing on web server logs for generalized association rules mining algorithm. *World Academy of Science, Engineering and Technology 48 2008*, pages 189–197, 2008.

M. R. Hess and J. D. Kromrey. Robust confidence intervals for effect sizes: A comparative study of cohens d and cliffs delta under nonnormality and heterogeneous variances. In *Proceedings of annual meeting of the American Educational Research Association*. San Diego, April 2004.

T. Hesterberg, S. Monaghan, D.S. Moore, A. Clipson, , and R. Epstein. *The Practice of Business Statistics*. W. H. Freeman and Company, 2003.

D.V. Hinkley. On the ratio of two correlated normal random variables. *Biometrika*, 56 (3):635–639, 1969.

L.V. Hodges and I. Olkin. *Statistical Methods for Meta-Analysis*. Academic Press, 1985.

E. C. Holmgren. The p-p plot as a method of comparing treatment effects. *Journal of the American Statistical Association*, 90(429):360–365, 1995.

D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, 1989.

Jr. Hosmer, D.W and S. Lemeshow. *Applied Logistic Regression*. New York: John Wiley & Sons, second edition, 2000.

B. Huberman, P. Pirolli, J. Pitkow, and R. Lukose. Strong regularities in world wide web surfing. *Science*, 97:280–295, 1997.

M. Hubert and E. Vandervieren. An adjusted boxplot for skewed distributions. *Computational Statistics & Data Analysis*, 52(12):5186–5201, 2008.

S. Jackman. *Bayesian Analysis for the Social Science*. John Wiley and Sons, New York, 2009.

W. Jank and G. Shmueli. A special issue on statistical challenges and opportunities in electronic commerce research. *Statistical Science*, 21, 2006.

A. Jasra, C.C. Holmes, and D.A. Stephens. Markov chain monte carlo methods and the label switching problem in bayesian mixture modelling. *Statistical Science*, 20:50–67, 2005.

E. J. Johnson, W. W. Moe, P. S. Fader, S. Bellman, and G. L. Lohse. On the depth and dynamics of online search behavior. *Management Science*, 50(3):299–308, 2004.

N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous univariate distributions*, volume 1. New York: John Wiley and Sons, 2nd edition, 1994.

T.O. Jones and W.E. Jr. Sasser. Why satisfied customers defect. *Harvard Business Review*, pages 88–99, (NovemberDecember) 1995.

B.H. Juang and L.R. Rabiner. Hidden markov models for speech recgnition. *Technometrics*, 33:251–272, 1991.

R.E. Kass and L. Wasserman. The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91:1343–1370, 1996.

A. Kaushik. *Web Analytics: An hour a day*. Wiley, 2007.

A. Kaushik. *Web Analytics 2: The Art of Online Accountability and Scinece of Customer Centricity*. Wiley, 2010.

H.J. Keselman, C.J. Huberty, L.M. Lix, S. Olejnik, R.A. Cribbie, B. Donahue, R.K. Kowalchuk, L.L. Lowman, Keselman J.C. Petoskey, M.D., and J.R. Levin. Statistical practices of educational researchers: An analysis of their anova, manova, and ancova analyses. *Review of Educational Research*, 68(3):350–386, 1998.

R. Kimball and R. Merz. *The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse*. John Wiley & Sons, 2000.

J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7:373–397, 2003.

R. Kohavi, L. Mason, R. Parekh, and Z. Zheng. Lessons and challenges from mining retail e-commerce data. *Machine Learning*, 57:83–113, 2004.

P. Kolari and A. Joshi. Web mining: Research and practice. *Computing in Science & Engineering*, 6(4):49–53, 2004.

R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2: 1–15, 2000.

J. Krueger. Null hypothesis significance testing. on the survival of a flawed method. *American Psychologist*, 56(1):16–26, 2001.

E. Kulinskaya and R. G. Staudte. Interval estimates of weighted effect sizes in the one-way heteroscedastic anova. *British Journal of Mathematical and Statistical Psychology*, 59: 97–111, 2006.

S. Kullback. *Information Theory and Statistics*. Publications Inc., Mineola, New York, 1968.

R.D. Ledesma, G. Macbeth, and N. Cortada de Kohan. Computing effect size measures with vista the visual statistics system. *Tutorials in Quantitative Methods for Psychology*, 5(1):25–34, 2009.

R.V. Lenth. Some practical guidelines for effective sample-size determination, 2001.

G. Li, R.C. Tiwari, and M.T. Wells. Quantile comparison functions in two-sample problems, with application to comparisons of diagnostic markers. *Journal of the American Statistical Association*, 91(434):689–698, 1996.

S. Li, J.C. Liechty, and A.L. Montgomery. Modeling category viewership of web users with multivariate count models. In *Statistics in an Era of Technological Change (JSM 2002)*, Joint Statistical Meetings, New York, USA, 2002.

B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. Springer, 2006.

Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer-Verlag, 2007.

P. Mair and M. Hudec. Session clustering using mixtures of proportional hazards models. Research Report Series 63, Department of Statistics and Mathematics Wirtschaftsuniversität Wien, March 2008. URL `http://statmath.wu-wien.ac.at/`.

H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Statistics*, 18:50–60, 1947.

N. R. Mann, R. E. Schafer, and N. D. Singpurwalla. *Methods for statistical analysis of reliability and life data*. John Wiley and Sons, New York, 1974.

Z. Markov and D.T. Larose. *Data Mining the Web: Uncovering Pattern in Web Content, Structure, and Usage*. Jonh Wiley and Sons, 2007.

P. McCullagh and J.A. Nelder. *Generalized Linear Models*. London: Chapman Hall, 1989.

K. McGraw and S. Wong. A common language effect size statistic. *Psychological Bulletin*, 111:361–365, 1992.

G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

G.J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

B. Mobasher. *Encyclopedia of Data Warehousing and Mining*, chapter Web Usage Mining. Idea Group, 2006.

B. Mobasher. Web personalization. *Lecture Notes in Computer Science*, 4321:557–575, 2007.

B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communication of ACM*, 43(8), 2000.

W. Moe. Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream. *Journal of Consumer Psychology*, 13(1):29–36, 2003.

W. Moe and P.S. Fader. Dynamic conversion behaviour at e-commerce sites. *Journal of Consumer Psychology*, 50:326–335, 2004.

W. W. Moe and P. S. Fader. Capturing evolving visit behavior in clickstream data. *Journal of Interactive Marketing*, 18:5–19, 2000.

W. W. Moe and P. S. Fader. Which visits lead to purchases? dynamic conversion behavior at e-commerce sites, 2001.

A. L. Montgomery. Applying quantitative marketing techniques to the internet. *Interfaces*, 31(2):90–108, 2001.

A. L. Montgomery, S. Li, K. Srinivasan, and J. C. Liechty. Modeling online browsing and path analysis using clickstream data. *Markeing Science*, 23(4):579–595, 2004.

F. Mosteller and J. W. Tukey. *Data Analysis and Regression: A Second Course in Statistics*. Reading, MA: Addison-Wesley, 1977.

R. Muller and M. Möckel. Logistic regression and cart in the analysis of multimarker studies. *Clinica Chimica Acta*, 394:1–6, 2008.

N. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

Olfa Nasraoui. *Encyclopedia of Data Mining and Data Warehousing*, chapter World Wide Web Personalization. Idea Group, 2005.

K. Natheer and C. C. Chan. Active user-based and ontology-based web log data preprocessing for web usage mining. Proceedings of theIEEE/WIC/ACM International Conference on Web Intelligence (WI'06), 2006.

B. Padmanabhan, Z. Zheng, and S.O. Kimbrough. Personalization from incomplete data: What you dont know can hurt. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001.

Y. H. Park and P. S. Fader. Modeling browsing behavior at multiple websites. *Marketing Science*, 23:280–303, 2004.

E. T. Peterson. *Web analytics demystified: a marketer's guide to understanding how your web site affects your business*. Celilo Group Media, 2004.

P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the web. In *Conference on Human Factors in Computing Systems (CHI-96)*, Vancouver, British Columbia, Canada, 1996.

Kai Puolamki and Samuel Kaski. Bayesian solutions to the label switching problem. In Niall Adams, Cline Robardet, Arno Siebes, and Jean-Franois Boulicaut, editors, *Advances in Intelligent Data Analysis VIII*, volume 5772 of *Lecture Notes in Computer Science*, pages 381–392. Springer Berlin / Heidelberg, 2009.

Y. Qi, J. W. Paisley, and L. Carin. Music analysis using hidden markov mixture models, 2007.

L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. volume 77, pages 257–286. IEEE, 1989.

C. R. Rao. *Linear Statistical Inference and Its Applications, 2nd Edition.* John Wiley and Sons, New York, 1973.

R.A. Redner and H.F. Walker. Mixture densities, maximul likelihood and the em algorithm. *SIAM Review*, 26:195–239, 1984.

S. Richardson and P.J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society, B*, 59:731–792, 1997.

C. P. Robert and D . M. Titterington. Reparametrization strategies for hidden markov models and bayesian approaches to maximum likelihood estimation. *Statistics and Computing*, 8:145–158, 1998.

C.P. Robert, G. Celeux, and J. Diebolt. Bayesian estimation of hidden markov chains: a stochastic implementation. *Statistics and Probability letters*, 16(1):77–83, 1993.

Rydén T. Robert, C. P. and D. M. Titterington. Bayesian inference in hidden markov models through the reversible jump markov chain monte carlo method. *Journal of the Royal Statistical Society, B*, 62:57–75, 2000.

D. Roussinov and L. Zhao. Automatic discovery of similarity relationships through web mining. *Decision Support Systems*, 35:149–166, 2003.

R. R. Sarukkai. Link prediction and path analysis using markov chains. In *Proceedings of the 9th Internationl World Wide Web Conference (WWW00)*, pages 377–386, 2000.

A. Schliep, C. Steinhoff, and A. Schonhuth. Robust inference of groups in gene expression time-courses using mixtures of hmms. *Bioinformatics*, 20:283289, 2004.

D.C. Schmittlein and R.A. Peterson. Customer base analysis: An industrial purchase process application. *Marketing Science*, 13(1):41–67, 1994.

G. E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

S.L. Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of American Statistical Association*, 97:337–351, 2002.

S.L. Scott and I. Hann. A nested hidden markov model for internet browsing, 2007.

S.L. Scott and P. Smyth. The markov modulated poisson process and markov poisson cascade with applications to web traffic modeling. In J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, and M. West, editors, *Bayesian Statistics*, volume 7, pages 671–680. Oxford University Press, 2003.

C. Sismeiro and R. E. Bucklin. Modeling purchase behavior at an e-commerce web site: A task-completion approach. *Journal of Marketing Research*, 41(3):306–232, 2004.

A.F.M. Smith and G.O. Roberts. Bayesian computation via the gibbs sampler and related markov chains. *Journal of Royal Statistical Society, B*, 55:167–174, 1993.

P. Smyth. Clustering sequences with hidden markov models. In *Advances in Neural Information Processing Systems*, pages 648–654. MIT Press, 1997.

M. Sperrin, T. Jaki, and E. Wit. Probabilistic relabelling strategies for the label switching problem in bayesian mixture models. *Stat Comput*, 20:357–366, 2010.

M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. A framework for the evaluation of session reconstruction heuristics in web usage analysis. *INFORMS Journal on Computing*, 15(2):171–190, 2003.

M. Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Series B, Methodological*, 62:795–810, 2000.

J. Sterne. *Web Metrics: Proven Methods for Measuring Web Site Success*. John Wiley & Sons, 2002.

Pang-ning Tan and Vipin Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, 6:9–35, 2002.

D. Tanasa and B. Trousse. Advanced data preprocessing for intersite web usage mining. *IEEE Intelligent Systems*, 19(2):59–65, 2004.

M. Tanner. *Tools for Statistical Inference*. Springer, New York, 1991.

B. Thompson. Statistical significance and effect size reporting: Portrait of a possible future. *Research in the Schools*, 5(2):33–38, 1998.

Smith A. F. M. Titterington, D. M. and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.

J. W. Tukey. *A survey of sampling from contaminated normal distributions In I. Olkin et al. (Eds.) Contributions to Probability and Statistics.* HStanford, CA: Stanford University Press, 1960.

D. Van den Poel and W. Buckinx. Predicting online-purchasing behavior. *European Journal of Operational Research*, 166(2):557–575, 2005.

W. Weibull. A statistical distribution function of wide applicability. *ASME Journal of Applied Mechanic, Transactions of the American Society of Mechanical Engineers*, 18 (3):293–297, September 1951.

Vic Werner, Craig Abramson, and Kenny Kistler. E-business and the corporate information factory. *Journal of Data warehousing*, 7(2):21–27, 2002.

H. Werthner, H.R. Hansen, and F. Ricci. Ieee computer society. In *In Proceedings of the Seventh International World Wide Web Conference*, 2007.

M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 1997.

H. White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.

R. R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing*. Elsevier Academic Press, 2005.

R. R. Wilcox and T. S. Tian. Measuring effect size: A robust heteroscedastic approach for two or more groups. *in press*, 2011.

M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55:1–17, 1968.

C.R. Wilson Van Voorhis and B.L. Morgan. Understanding power and rules of thumb for determining sample sizes. *Tutorials in Quantitative Methods for Psychology*, 3(2): 43–50, 2007.

J.M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. MIT Press, 2010.

C. Wu and H.L. Chen. Counting your customers: Compounding customers in-store decisions, interpurchase time and repurchasing behaviour. *European Journal of Operational Research*, 127(1):109–119, 2000.

A. Ypma and T. Heskes. Categorization of web pages and user clustering with mixtures of hidden markov models. In *Workshop on WebKDD-2002*, pages 35–49, 2002.

Appendix A

Robust Effect Sizes: E-commerce application

A.1 Algorithms for computing Effect Size

Algorithm 1 Calculate the empirical quantile function, **eqf**($x[\,]$,$p[\,]$)

Require: a vector of observations $x[\,]$ based on which empirical quantile function is computed
Require: a vector of probabilities $p[\,]$ for which the empirical quantile function is computed

$n_x \leftarrow$ length of the vector $x[\,]$
$n_p \leftarrow$ length of the vector $p[\,]$
for $i = 1$ to n_x **do**
    $F_x[i] \leftarrow i/n_x$
end for
for $i = 1$ to n_p **do**
    $q_x[i] \leftarrow$ minimum value of $x[\,]$
end for
for $j = 1$ to n_p **do**
    for $i = 1$ to $n_x - 1$ **do**
        if $p[j] \geq F_x[i]$ **then**
            $q_x[j] \leftarrow x[i+1]$
        end if
    end for
end for
return $q_x[\,]$

Algorithm 2 Calculate the empirical QAD effect size, **qad**$(x[\,], y[\,])$

Require: a vector of observations $x[\,]$
Require: a vector of observations $y[\,]$
$x[\,] \leftarrow$ sort vector of $x[\,]$ by ascending order
$y[\,] \leftarrow$ sort vector of $y[\,]$ by ascending order
$N_x \leftarrow$ length of the vector $x[\,]$
$N_y \leftarrow$ length of the vector $y[\,]$
for $i = 1$ to n_x **do**
    $F_x[i] \leftarrow i/n_x$
end for
for $i = 1$ to n_y **do**
    $F_y[i] \leftarrow i/n_y$
end for
$p[\,] \leftarrow$ merge F_x and F_y and sort them by ascending order
$n_p \leftarrow$ length of the vector $p[\,]$
$Q_x \leftarrow$ **eqf**$(x[\,], p[\,])$
$Q_y \leftarrow$ **eqf**$(y[\,], p[\,])$
for $i = 1$ to n_p-1 **do**
    $d_p[i] \leftarrow p[i+1] - p[i]$
end for
for $i = 2$ to n_p **do**
    $d_Q[i-1] \leftarrow \left|Q_x[i] - Q_y[i]\right|$
end for
$qad \leftarrow 0$
for $i = 2$ to n_p-1 **do**
    $qad \leftarrow qad + d_Q[i] \times d_p[i])$
end for
return qad

Algorithm 3 Calculate the empirical cdf, **ecdf**$(x[\,],\ q[\,])$

Require: a vector of observations $x[\,]$ based on which the cdf is computed
Require: a vector of quantiles $q[\,]$ for which the cdf is computed
$x[\,] \leftarrow$ sort the vector of $x[\,]$ by ascending order
$n_x \leftarrow$ length of the vector $x[\,]$
$q[\,] \leftarrow$ sort the vector of $q[\,]$ and keep unique values
$n_q \leftarrow$ length of the vector $q[\,]$
for $i = 1$ to n_q **do**
    $sum \leftarrow 0$
    for $j = 1$ to n_q **do**
        if $x[j] \leq q[i]$ **then**
            $sum \leftarrow sum + 1$
        end if
    end for
    $F[i] \leftarrow sum/n_x$
end for
return $F[\,]$

Algorithm 4 Calculate the area of non-self-intersecting polygon with n vertices, **area**$(v[\,,\,])$

Require: a $(n+1) \times 2$ matrix of vertices $v[\,,\,]$ of a non-self-intersecting polygon

$area \leftarrow 0$
$n_v \leftarrow$ the number of vertices
for $i = 1$ to n_v **do**
    $area \leftarrow area + \frac{1}{2} \times \big(v[i,1] \times v[i+1,2] - v[i,2] \times v[i+1,1]\big)$
end for
return $area$

Algorithm 5 Calculate the empirical divergence measure, **div**$(x[\,], y[\,])$

Require: a vector of observations $x[\,]$
Require: a vector of observations $y[\,]$

$x[\,] \leftarrow$ sort the vector of $x[\,]$ by ascending order
$GGinv \leftarrow [0, \textbf{ecdf}(x[\,], x[\,]), 1]$
$FGinv \leftarrow [0, \textbf{ecdf}(y[\,], x[\,]), 1]$
$n \leftarrow$ length of GGinv
for $i = 1$ to n **do**
    $v[i,1] \leftarrow GGinv[i]$
end for
for $i = 1$ to n **do**
    $v[i,2] \leftarrow FGinv[i]$
end for
for $j = 1$ to 2 **do**
    $v[n+1,j] \leftarrow 0$
end for
for $i = 1$ to $n+1$ **do**
    if $v[i,1] > v[i,2]$ **then**
        $tmp \leftarrow v[i,1]$
        $v[i,1] \leftarrow v[i,2]$
        $v[i,2] \leftarrow tmp$
    end if
end for
$div \leftarrow 2 \times$ **area**$(v[\,,\,])$
return div

Algorithm 6 Calculate the empirical quantile comparison effect size, **qces**$(x[\,], y[\,])$

Require: a vector of observations $x[\,]$
Require: a vector of observations $y[\,]$

$div_{xy} \leftarrow$ **div**$(x[\,], y[\,])$
$div_{yx} \leftarrow$ **div**$(y[\,], x[\,])$
$qces \leftarrow \frac{1}{2} \times div_{xy} + \frac{1}{2} \times div_{yx}$
return $qces$

Appendix B

R Codes

B.1 Plots and Exploratory Analysis

```
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Produce a heat-plot based on three numeric vector
#
#-------------------------------------------------------------------------------
heat.plot <- function(x, y, z, seg=100,
                      xlabt=NULL, ylabt=NULL,
                      maxx=NULL, maxy=NULL,cex=3)
   {

  if(is.null(maxx)) maxx <- max(x)
  if(is.null(maxy)) maxy <- max(y)

xd <- (maxx-min(x))/seg
yd <- (maxy-min(y))/seg
brkx <- seq(min(x),maxx, xd)
brky <- seq(min(y),maxy, yd)
brkx[1]<- min(x)-0.000001
brky[1]<- min(y)-0.000001

c <- length(brkx)
x.mid <- (brkx[1:c-1]+brkx[2:c])/2
y.mid <- (brky[1:c-1]+brky[2:c])/2
x.cat <- cut(x=x, breaks=brkx, right=T)
y.cat <- cut(x=y,  breaks=brky, right=T)

tot <- table(x.cat,y.cat)
den<- tot
den[tot==0]<-1
tab   <- table(x.cat,y.cat, z)
tab   <- tab[,,2]
pct   <- (tab/den)*100
  pct[tot<10]<-tab[tot<10]
```

```
tab <- round(pct,0)
freq <- as.vector(tab)
maxt <- 100

heat.col <- heat.colors(maxt)

colv <- heat.col[maxt-freq]
colv[as.vector(tot==0)]<-0
x.points <- rep(x.mid, times=seg)
y.points <- rep(y.mid, each= seg)

if (is.null(xlabt)) xlabt<- deparse(substitute(x))
if (is.null(ylabt)) ylabt<- deparse(substitute(y))

  nf <- layout(matrix(c(1,2), 1,2), c(6,1), TRUE)
  layout.show(nf)

  par(mar=c(5,5,2,1))
plot(x.points,
     y.points,
     col=colv,
     pch=15,
     cex=cex,
     xlab=xlabt,
     ylab=ylabt,
     cex.lab=1.1,
       cex.axis=1.1)
  box()

  par(mar=c(5,0,2,3))
  image(matrix(1:100,1), col=heat.colors(100)[100:1], axes=F, xlim=c(0,.1))
  axis(4,at=seq(0,1,length.out=11),labels=seq(0,100,10), cex.axis=1.1)
  box()
  layout(matrix(c(1)))
}


#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Produce a interaction-plot for representing inter-action effect in
#   logistic regression
#-------------------------------------------------------------------------------
interact.curve<-
    function(depend,
     independ,
     class,
     xlabt="Explanatory variable",
     ylabt="Probability of Conversion",
     maxt=0,
     leg.txt=c("0","1"),
     dit=TRUE, scat=TRUE, bounds=TRUE, curve=TRUE)
  {
  if (maxt==0) maxt <- max(independ)

  # ------------------------- drawing a empty plot to put shapes in
  plot(independ, depend,
```

```
type="n",
family="serif",
xlab=xlabt,
ylab=ylabt,
cex.lab=1.2,
xlim=c(min(independ),maxt),
ylim=c(-0.2,1.2)
)

  logi.scat <- function(depend, independ, colt=3)
    {
    model <- glm(depend~ independ,
  family=binomial(link=logit))
    qnt<- quantile(independ,seq(0,1,0.01))
    qnt <- unique(qnt)
    c<- length(qnt)
    indep.mid <- (qnt[1:c-1]+qnt[2:c])/2
    qnt[1]<- min(independ)-0.000001
    indep.cat<-cut(x=independ,breaks=qnt, right=T, labels=indep.mid)
    #--------------- make table to find the percent of conversion
    tab<-table(depend,indep.cat)
    pct=(tab[2,]/(tab[1,]+tab[2,]))
    #------------- put the percent point on the plot
    points(indep.mid,pct, pch=20, cex=0.8 , col=colt )
    }

# ------ add fitted values by simple logistic model
  logi.curve <- function(depend, independ, colt=2)
  {
  model <- glm(depend~ independ,
  family=binomial(link=logit))
  new.indep  <- seq(min(independ),maxt,(maxt-min(independ))/100)
  prob <- predict(model,
  data.frame(independ=new.indep),
  type="response")
  lines(new.indep, prob , col=colt, lwd=1.5, lty=1)
  }

  logi.dit <-
      function (depend,independ, inside=TRUE, colt=-1)
  {
  indep.0 <- independ[depend == 0]
  indep.1 <- independ[depend == 1]

  n  <- length(depend)
  n0 <- length(indep.0)
  n1 <- length(indep.1)

  uni.plot.0 <- function(x) length(which(indep.0 == x))/length(indep.0)
  uni.plot.1 <- function(x) length(which(indep.1 == x))/length(indep.1)

  # get the number of repeated values of "independ":

  uni.indep.0 <- unique(indep.0)
  uni.indep.1 <- unique(indep.1)

  r0 <- rank(uni.indep.0)
```

```
  r1 <- rank(uni.indep.1)

  uni.indep.0[r0]<-uni.indep.0
  uni.indep.1[r1]<-uni.indep.1

  cosa.0 <- apply(as.matrix(unique(indep.0)), 1, uni.plot.0)
  cosa.1 <- apply(as.matrix(unique(indep.1)), 1, uni.plot.1)

  scale <- 0.2
  s.cosa.0 <- cosa.0/max(cosa.0)*scale
  s.cosa.1 <- cosa.1/max(cosa.1)*scale

  s.cosa.0 [r0]<- s.cosa.0
  s.cosa.1 [r1]<- s.cosa.1

  c0 <- length(uni.indep.0)
  c1 <- length(uni.indep.1)

  if (inside)
  {
  x0 <- uni.indep.0
  y0 <- rep(0, length(uni.indep.0))
  x1 <- c(uni.indep.0[2:c0],maxt)
  y1 <- s.cosa.0

  w0 <- uni.indep.1
  z0 <- 1-s.cosa.1
  w1 <- c(uni.indep.1[2:c1],maxt)
  z1 <- rep(1, length(uni.indep.1))
  }

  if (!inside)
  {
  x0 <- uni.indep.0
  y0 <- rep(0, length(uni.indep.0))
  x1 <- c(uni.indep.0[2:c0],maxt)
  y1 <- -s.cosa.0

  w0 <- uni.indep.1
  z0 <- 1+s.cosa.1
  w1 <- c(uni.indep.1[2:c1],maxt)
  z1 <- rep(1, length(uni.indep.1))
  }


rect(x0,y0,x1,y1, lwd=1, col=colt)
rect(w0,z0,w1,z1, lwd=1, col=colt)
}


dep0 <- depend[class==0]
indep0 <- independ[class==0]
logi.scat(dep0, indep0,  col=1)
logi.curve(dep0, indep0, col=1)
logi.dit(dep0, indep0,inside=TRUE)
```

```
dep1 <- depend[class==1]
indep1 <- independ[class==1]
logi.scat(dep1, indep1,  col=2)
logi.curve(dep1, indep1, col=2)
logi.dit(dep1, indep1, inside=FALSE, colt=2)

legend(legend=leg.txt, "topright",
col = c(1,2),
text.col = "black",
lty= rep(1,2),
pch= rep(20,2),
merge = TRUE)

}

#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Produce ROC-curve for the fitted logistic regression model
#
#-------------------------------------------------------------------------------


ROC.logi <- function(plott=FALSE)
    {

    n <- nrow(wdata)
    set.seed(17)
    train <- sample(1:n, 7000)
    model <- glm(formula=stepwise.model$formula,
  data=wdata[train,],
  family=binomial(link=logit))
    test.data<- wdata[-train,]
    pred <- predict(model,
    newdata=test.data,
    type="response")

    pct1<-table(wdata$cvs)/nrow(wdata)
    cutpoints <- (0:20)/20
    spec <- sens <- misspec <- phi <- NULL
    for (i in cutpoints)
       {
       class<- cut(pred,c(-Inf,i,Inf),label=c(0,1))
       tab <- table(class, wdata$cvs[-train])

       T0  <- tab[1,1]
       T1  <- tab[2,2]
       F1  <- tab[1,2]
       F0  <- tab[2,2]

       spec <-     cbind(spec,   T0/(tab[1,1]+tab[2,1]))
       sens <-     cbind(sens,   T1/(tab[1,2]+tab[2,2]))
       misspec <-  cbind(misspec, 1-T0/(tab[1,1]+tab[2,1]))
       phi <-      cbind(phi,    (T0+T1)/sum(tab))
       }

plott= TRUE
```

```
    if(plott)
  {
  plot(misspec, sens,
type="b",
pch = 15,
axes=FALSE,
ylim=c(0,1),
xlim=c(0,1),
ylab="Sensitivity",
xlab="1-Specificity",
cex.lab=1.2,
lwd=2,
    cex.lab=1.2,
    cex.axis=1.2,
    font.lab=1,
    col="red"
    )

    abline(0,1, lwd=2)

   axis(1,seq(0,1,0.1), cex.axis=1.1)
 axis(2,seq(0,1,0.1), cex.axis=1.1)
 box()
 grid(nx=NULL,ny=NULL, col="black")
 text(x=0.7,y=0.42, "Estimated area = 0.849", cex=1.2)

 }
     invisible(list(spec,sens, misspec,phi))
      }
ROC.logi(plott=TRUE)
```

B.2 Effect Size

```
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Compute QAD effect size for two Weibul distributions
#
#-------------------------------------------------------------------------------

qad = function(shape0,scale0, shape1, scale1)
    {
    incr = 1e-4
    p =  seq(0,1,incr)
    u0 <- qweibull(p,shape0, scale0)
    u1 <- qweibull(p,shape1, scale1)
    qad = sum(abs(incr*abs(u1-u0)[-1]))
    return(qad)
    }

#-------------------------------------------------------------------------------
```

```
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Compute QC effect size for two Weibul distributions
#-------------------------------------------------------------------------------

qces = function(shape0,scale0, shape1, scale1)
    {
    incr = 1e-4
    p =  seq(0,1,incr)
    u = pweibull(qweibull(p,shape1, scale1), shape0, scale0)
    w = pweibull(qweibull(p,shape0, scale0), shape1, scale1)
    qces = sum(abs(incr*abs(u-p)[-1]))+sum(abs(incr*abs(w-p)[-1]))
    return(qces)
    }

#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Compute Cliff's delta effect size for two arbitrary distributions
#
#-------------------------------------------------------------------------------

cliff = function(y0,y1)
      {
      n1 <- length(y0)
      n2 <- length(y1)
      sgn <- numeric(n1*n2)
      for (i in 1:n1)
     for (j in 1:n2)
        sgn[(i-1)*n2+j] <- sign(y0[i]-y1[j])
      d <- sum(sgn)/(n1*n2)
      return(d)
      }

#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Compute Cohen's d effect size for two arbitrary distributions
#
#-------------------------------------------------------------------------------

cdes = function(y0,y1)
      {
      n0 <- length(y0)
      n1 <- length(y1)
      mu0 <- mean(y0)
      mu1 <- mean(y1)
      sp <- ((n0-1)*var(y0)+(n1-1)*var(y1))/(n1+n0-2)
      d  <- (mu1-mu0)/sqrt(sp)
      return(d)
      }

#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Compute Empirical QAD effect size for two arbitrary distributions
#
```

```
#-------------------------------------------------------------------------------

eqad = function(x, y, plot=TRUE)
    {
    y    = sort(y)
    x    = sort(x)
    ny   = length(y)
    nx   = length(x)
    My.eqf = function(x,p)
  {
  nx = length(x)
  Fx = (1:nx)/nx
  np = length(p)
  Qx = rep(min(x),np)
  for (j in 1:np)
  for (i in 1:(nx-1))
  if  (p[j] >= Fx[i])
      {
      Qx[j] = x[i+1]
      }
  return(Qx)
  }

    p = sort(unique(c((0:nx)/nx, (0:ny/ny))))
    np = length(p)
    Qx = My.eqf(x,p)
    Qy = My.eqf(y,p)

    diff.p = diff(p)
    diff.Q = abs(Qx-Qy)[-1]
    eqad = sum(diff.Q*diff.p)
    return(eqad)
    }

plotqad = function(x, y)
    {
    y    = sort(y)
    x    = sort(x)
    ny   = length(y)
    nx   = length(x)
    eqf.point = function(x)
{
nx = length(x)
ux = unique(x)
p = numeric(length(ux))
for (i in 1:length(ux))
      p[i] = (1/nx)*sum(x<=ux[i])
foo1= c(0,rep(p,each=2))
foo1 = foo1[-length(foo1)]
foo2= rep(ux,each=2)
mat=cbind(foo1,foo2)
return(mat)
}
    foox=eqf.point(x)
    fooy=eqf.point(y)

    mat=rbind(foox,fooy[nrow(fooy):1,],foox[1,])
```

```
    plot(foox, type='l', lwd=1.5, xlim=c(0,1), ylim=c(min(x,y),max(x,y)))
    polygon(mat,col="lightgreen",border="white")
    lines(foox, type='l', lwd=1.5, xlim=c(0,1), ylim=c(0,10))
    lines(fooy, type='l', lty=2, lwd=1.5)
    }

#-----------------------------------------------------------------------------
#-----------------------------------------------------------------------------
#-----------------------------------------------------------------------------
#   Compute Empirical QC effect size for two arbitrary distributions
#
#-----------------------------------------------------------------------------

eqces = function(x, y, plot=TRUE)
    {
    y    = sort(y)
    x    = sort(x)
    ny   = length(y)
    nx   = length(x)

    My.area = function(n)
{
area = 0
for (i in 1:(nrow(n)-1))
    area = area + 0.5 * (n[i,1]*n[i+1,2]-n[i,2]*n[i+1,1])
return(area)
}

    My.ecdf = function(x,q)
    {
    x  = sort(x)
    nx = length(x)
    q  = sort(unique(q))
    F  = numeric(length(q))
    for (i in 1:length(q))
    F[i]  = (1/nx)*sum(x<=q[i])
    return(F)
    }

    GGinv = c(0, My.ecdf(x,x), 1)
    FGinv = c(0, My.ecdf(y,x), 1)
    m = cbind(GGinv,FGinv)
    m = rbind(m,c(0,0))
    if (plot==TRUE)
 {
plot(m, type='l')
polygon(m,col="lightgreen",border="darkgreen")
lines(m, type='l')
grid(nx=NULL, lwd=1.5, col="grey")
}
    n = m
    n[n[,1]<=n[,2],] = n[n[,1]<=n[,2],2:1]
    eqces = My.area(n)
    names(eqces) = NULL
    return(eqces)
    }
```

B.3 Mixture of Hidden Markov Model

```
##-----------------------------------------------------------------------------
##---------------------------- Generate Mixture of hidden Markov Model

sim.mhmm = function(I, seed, omega, tpm, epm)
      {
      ispd = list(NULL)
      K= length(omega)

      for (k in 1:K)
         ispd[[k]] = revise.ispd(tpm[[k]])

      set.seed(seed)
      S = nrow(tpm[[1]])
      L = nrow(epm[[1]])
      Ci= rdisc(I, prob=omega)
      Ti   =  sample(30:120,I,TRUE)
      St   = list(NULL); for (i in 1:I) St[[i]] = numeric(Ti[i])
      Yt   = list(NULL); for (i in 1:I) Yt[[i]] = numeric(Ti[i])
      for (i in 1:I)
      for (t in 1:Ti[i])
        {
        if (t==1) St[[i]][t]=rdisc(1,ispd[[Ci[i]]])
        if (t>1)  St[[i]][t]=rdisc(1,tpm[[Ci[i]]][St[[i]][t-1],])
        Yt[[i]][t]=rdisc(1,epm[[Ci[i]]][,St[[i]][t]])
        }
      return(list(Yt=Yt,St=St, Ci=Ci))
      }

#-----------------------------------------------------------------------------
#-----------------------------------------------------------------------------
#-----------------------------------------------------------------------------
#   Perform mixture of Hidden Markov Model on matrix of discrete observations
#   Using EM Algorithm
#-----------------------------------------------------------------------------

My.mhmm <-
function(y, yval = NULL, par0 = NULL, M = NULL, K=NULL, rand.start = NULL,
         tolerance = 1e-04, verbose = FALSE, itmax = 200,  tying=FALSE,
         crit = "Linf", data.name = NULL)
    {
    if (is.null(data.name))
        data.name <- deparse(substitute(y))
    if (is.vector(y)) y <- matrix(y, ncol=1)
    y <- My.tidyup(y, yval)
    if (is.null(yval))
        yval <- sort(unique(as.vector(y)))
    OL <- length(yval)
    L  <- ncol(y)
    O  <- length(unique(as.vector(y)))
    if (is.null(par0) & is.null(M))
        stop("One of par0 and M must be specified.")
    cpd <- rep(1/K, K)
```

```
if (is.null(rand.start))
    rand.start <- list(tpm = TRUE, Rho = TRUE, cmp=TRUE)
parm  <- My.init.all(par0, L=L, M=M, K=K, OL=OL, rand.start)
icrit <- match(crit, c("PCLL", "L2", "Linf"))
if (is.na(icrit))
    stop("Stopping criterion not recognized.")
if (is.null(par0$cmp) & is.null(K))
    stop("One of par0$cmp and K must be specified.")
if (!is.null(par0$cmp))
    K <- ncol(par0$cmp)
old.theta <- NULL
new.theta <- NULL
for (k in 1:K)
     old.theta <- cbind(old.theta, c( c(parm[[k]]$tpm[, -M]),
                                      c(parm[[k]]$Rho[1:(m - 1),])
                                    )
                        )
old.ll <- -Inf
digits <- 2 + ceiling(abs(log10(tolerance)))

model <- list()
em.step <- 1
if (verbose)
    cat("\n      Initial set-up completed ...\n\n")
if (verbose)
    cat("Repeating ...\n\n")
chnge <- numeric(3)

repeat{
      if (verbose)
          cat(paste("EM step ", em.step, ":\n", sep = ""))
      for (k in 1:K)
          {
          model[[k]] <- My.recurse(y, tpm  = parm[[k]]$tpm,
                                      Rho  = parm[[k]]$Rho,
                                      ispd = parm[[k]]$ispd,
                                      cmp  = parm[[k]]$cmp
                                      )
          }
      cmp <- revise.cmp(model, cpd)
      cpd <- revise.cpd(cmp)
      ll  <- revise.ll(model, cpd)
      if (tying)
          {
          Rho <- 0
          for (k in 1:K)
              Rho = Rho + model[[k]]$Rho*cpd[k]
          for (k in 1:K)
              model[[k]]$Rho = Rho
          }
      for (k in 1:K)
          parm[[k]] <- list( tpm = model[[k]]$tpm,
                             ispd = model[[k]]$ispd,
                             Rho  = model[[k]]$Rho,
                             cmp  = cmp[,k])
      new.theta <- NULL
      for (k in 1:K)
```

```
            new.theta <- cbind(new.theta, c( c(parm[[k]]$tpm[, -M]),
                                             c(parm[[k]]$Rho[1:(m - 1),])
                                           )
                               )

        chnge[1] <- if (old.ll > -Inf)
            100 * (ll - old.ll)/abs(old.ll)  else Inf
        chnge[2] <- sqrt(sum((old.theta - new.theta)^2))
        chnge[3] <- max(abs(old.theta - new.theta))
        if (verbose) {
            cat("     Log-likelihood: ", format(round(ll, digits)),
                "\n", sep = "")
            cat("     Percent decrease in log-likelihood: ",
                format(round(chnge[1], digits)), "\n", sep = "")
            cat("     Root-SS of change in coef.: ", format(round(chnge[2],
                digits)), "\n", sep = "")
            cat("     Max. abs. change in coef.: ", format(round(chnge[3],
                digits)), "\n", sep = "" )
            }
        if (chnge[icrit] < tolerance) {
            theta <- new.theta
            converged <- TRUE
            nstep <- em.step
            break
        }
        if (em.step >= itmax) {
            cat("Failed to converge in ", itmax, " EM steps.\n",
                sep = "")
            theta <- new.theta
            converged <- FALSE
            nstep <- em.step
            break
        }
        old.theta <- new.theta
        old.ll <- ll
        em.step <- em.step + 1
    }
    ll <- revise.ll(model, cpd)
    names(model) <- paste("Cluster", 1:K, sep="")
    for (k in 1:K)
          model[[k]]$like.y <- NULL
    return(list(model = model,  cmp = cmp, cpd = cpd, log.like = ll, converged = converged,
        nstep = em.step, data.name = data.name))


#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#   Perform the Viterbi Algorithm for the mixture of Hidden Markov Model on
#   matrix of discrete observations
#-------------------------------------------------------------------------------

My.viterbi <-
function (y, object = NULL, tpm, Rho, ispd, yval = NULL)
{
    y <- tidyup(y, yval)
    n <- nrow(y)
```

```
    nc <- ncol(y)
    if (!is.null(object)) {
        tpm <- object$tpm
        Rho <- object$Rho
        ispd <- object$ispd
    }
    K <- nrow(tpm)
    rslt <- list()
    for (j in 1:nc) {
        psi <- list()
        delta <- ispd * Rho[y[1, j], ]
        delta <- delta/sum(delta)
        for (tt in 2:n) {
            tmp <- apply(delta * tpm, 2, function(x) {
                ((1:length(x))[x == max(x)])
            })
            psi[[tt]] <- tmp
            delta <- Rho[y[tt, j], ] * apply(delta * tpm, 2,
                max)
        }
        temp <- list()
        temp[[n]] <- (1:K)[delta == max(delta)]
        for (tt in (n - 1):1) {
            i <- 0
            temp[[tt]] <- list()
            for (x in temp[[tt + 1]]) {
                k <- x[1]
                for (w in psi[[tt + 1]][[k]]) {
                  i <- i + 1
                  temp[[tt]][[i]] <- c(w, x)
                }
            }
        }
        rrr <- matrix(unlist(temp[[1]]), nrow = n)
        rslt[[j]] <- if (ncol(rrr) == 1)
            as.vector(rrr)
        else rrr
    }
    if (nc == 1)
        rslt <- rslt[[1]]
    rslt
}
```

B.4 Bayesian Mixture of Hidden Markov Model

```
#-------------------------------------------------------------------------
#-------------------------- Invoke the necessary R packages
#--------------------------

invoke.lib = function()
```

```
    {
    library(MASS)
    library(lattice)
    library(coda)
    library(MCMCpack)
    library(hmm.discnp)
    }

#-------------------------------------------------------------------------------
#------------------------------- A function to generate general discrete values
#-------------------------------

rdisc =  function(n=1, prob=NULL, levels=NULL)
    {
    if (is.null(prob))
        {
        print("probability vector shuld be specified")
        stop
        }
    K =  length(prob)
    if (is.null(levels)) levels=1:K
    cum.prob=  cumsum(prob)
    foo =  rep(0, n)
    for (i in 1:n)
        {
        u =  runif(1)
        foo[i] =  levels[match(1,as.numeric(u <= cum.prob))]
        }
    return(foo)
    }

#-------------------------------------------------------------------------------
#----------------------------- A function to update initial state probabilities
#-----------------------------

revise.ispd =
function (tpm)
{
    eee =  eigen(t(tpm))
    k =  match(1, round(eee$values, 6))
    if (length(k) != 1)
     {
        cat("Problems with eigenvalues:\n")
        print(eee$values)
        stop()
     }
    ispd =  Re(eee$vectors[,k])
    ispd/sum(ispd)
}

#-------------------------------------------------------------------------------
#---------------------------- A function to help forward recursion computations
#-----------------------------

My.ffun =
function (y, Rho)
{
```

```
    M <- ncol(Rho)
    py <- Rho[y, 1:M]
    py[is.na(py)] <- 1
    return(py)
}

#------------------------------------------------------------------------------
#----------------------------- A function to put the initial values of the MCMC
#-------------------------------

get.init = function(tpm0=NULL, epm0=NULL, ispd0=NULL, St0=NULL, omega0=NULL, Ci0=NULL, mpm0=NULL)
    {
    if (is.null(tpm0)) {
 tpm0   =  list(NULL)
 for (k in 1:K)  tpm0[[k]] =  matrix(rdirichlet(S,alpha=delta), ncol=S)
}
assign("tpm", tpm0, envir=.GlobalEnv)

    if (is.null(epm0)) {
 epm0   =  list(NULL)
 for (k in 1:K)    epm0[[k]] =  t(rdirichlet(S,alpha=tau))
}
assign("epm", epm0, envir=.GlobalEnv)

    if (is.null(ispd0)) {
ispd0   =  list(NULL)
for (k in 1:K)    ispd0[[k]] =  revise.ispd(tpm[[k]])
}
assign("ispd", ispd0, envir=.GlobalEnv)


    if (is.null(omega0)) omega0 = rdirichlet(1,alpha=zeta)
assign("omega", omega0, envir=.GlobalEnv)

    if (is.null(St0))
     {
St0 = list(NULL)
for (i in 1:I)
for (t in 1:Ti[i])
     St0[[i]] =  ceiling(S*runif(Ti[i]))
}
assign("St", St0, envir=.GlobalEnv)

    if (is.null(mpm0))  {
mpm0 = matrix(0, nrow=I, ncol=K)
for (k in 1:K)
for (i in 1:I)
    mpm0[i,k] = sum(diag(log(epm[[k]][Yt[[i]],St[[i]]])))
mpm0 = exp(mpm0)/apply(exp(mpm0), MARGIN=1, FUN=sum)
assign("mpm", mpm0, envir=.GlobalEnv)
}

    if (is.null(Ci0))  Ci0 =  ceiling(K*runif(I))
assign("Ci", Ci0, envir=.GlobalEnv)
    }

#------------------------------------------------------------------------------
```

```
#------------------------------ Te starting loop for Gibbs sampler
#--------------------------------

hyper.par = function()
    {
    assign("delta", rep(1,S), envir=.GlobalEnv)
    assign("tau", rep(1,L), envir=.GlobalEnv)
    assign("zeta", rep(1,K), envir=.GlobalEnv)
    }

#-------------------------------------------------------------------------------
#------------------------------ Set vectors to keep the sampling outputs
#--------------------------------

set.vectors = function(n)
    {
    tpm.t   =  array(0, dim=c(Nr, K*S*S));    assign("tpm.t", tpm.t, envir=.GlobalEnv)
    epm.t   =  array(0, dim=c(Nr, K*S*L));    assign("epm.t", epm.t , envir=.GlobalEnv)
    ispd.t  =  array(0, dim=c(Nr, K*S))  ;    assign("ispd.t", ispd.t, envir=.GlobalEnv)
    omega.t =  array(0, dim=c(Nr, K))  ;    assign("omega.t", omega.t, envir=.GlobalEnv)
    Ci.t    =  array(0, dim=c(Nr, I))  ;    assign("Ci.t", Ci.t, envir=.GlobalEnv)
    llik.t  =  array(0, dim=c(Nr, 1))  ;    assign("llik.t", llik.t, envir=.GlobalEnv)
    St.t    =  array(0, dim=c(Nr, 10))  ;    assign("St.t", St.t, envir=.GlobalEnv)
    }

#-------------------------------------------------------------------------------
#------------------------------ Updare the transition matrix probabilities (tpm)
#---------------------------


update.tpm = function()
{
nij = list(NULL)
for (k in 1:K) nij[[k]] =  matrix(0, ncol=S, nrow=S)
for (i in 1:I)
for (t in 1:(Ti[i]-1))
nij[[Ci[i]]][St[[i]][t],St[[i]][t+1]] =  nij[[Ci[i]]][St[[i]][t],St[[i]][t+1]] + 1

for (k in 1:K)
for (s in 1:S) tpm[[k]][s,] = rdirichlet(1,alpha=nij[[k]][s,]+delta)
assign("tpm", tpm, envir=.GlobalEnv)
       return(tpm)
}

#-------------------------------------------------------------------------------
#------------------------------ Producing the probabilities and updating
#------------------------------ the hidden states by Direct Gibbs sampling
#--------------------------------
update.St.DG = function()
{
for (i in 1:I)
  {
  Pt =  matrix(0, nrow=Ti[i], ncol=S)
  Pt[1,] =  ispd[[Ci[i]]]*epm[[Ci[i]]][Yt[[i]][1],]*tpm[[Ci[i]]][,St[[i]][2]]
  Pt[1,] =  Pt[1,]/sum(Pt[1,])
  St[[i]][1] =   rdisc(n=1, prob=Pt[1,])
  for (t in 2:(Ti[i]-1))
```

```
{
Pt[t,] =  tpm[[Ci[i]]][St[[i]][t-1],]*epm[[Ci[i]]][Yt[[i]][t],]* tpm[[Ci[i]]][,St[[i]][t+1]]
Pt[t,] =  Pt[t,]/sum(Pt[t,])
St[[i]][t]  =  rdisc(n=1, prob=Pt[t,])
}
  Pt[Ti[i],] =  tpm[[Ci[i]]][St[[i]][Ti[i]-1],]*epm[[Ci[i]]][Yt[[i]][Ti[i]],]* rep(1,S)
  Pt[Ti[i],] =  Pt[Ti[i],]/sum(Pt[Ti[i],])
  St[[i]][Ti[i]] =   rdisc(n=1, prob=Pt[Ti[i],])
  }
assign("St", St, envir=.GlobalEnv)
return(St)
}

#-------------------------------------------------------------------------
#----------------------------- Producing the probabilities and updating
#----------------------------- the hidden states by
#----------------------------- Stochastic Forward-Backward Gibbs sampling

update.St.FB = function()
{
for (i in 1:I)
  {
    if (S>1)
{
P_t =  array(0,dim=c(S,S,Ti[i]))
Pt  =  matrix(0, nrow=Ti[i], ncol=S)
pij =  matrix(0, nrow=Ti[i], ncol=S)
pij[1,] =  ispd[[Ci[i]]] * epm[[Ci[i]]][Yt[[i]][1],]
pij[1,] =  pij[1,]/sum(pij[1,])
for (t in 2:Ti[i])
    {
    for (s in 1:S)
P_t[s,,t] =  pij[t-1,s]* epm[[Ci[i]]][Yt[[i]][t],]* tpm[[Ci[i]]][s,]
    P_t[,,t] =  P_t[,,t]/sum(P_t[,,t])
    pij[t,] =  apply(P_t[,,t], MARGIN=2, FUN=sum)
    }
Pt[Ti[i],] =  apply(P_t[,,Ti[i]], MARGIN=2,FUN=sum)
St[[i]][Ti[i]] =   rdisc(n=1, prob=Pt[Ti[i],])
for (t in 1:(Ti[i]-1))
      {
      Pt[Ti[i]-t,]      =  P_t[,St[[i]][Ti[i]-t+1],Ti[i]-t+1]/sum(P_t[,St[[i]][Ti[i]-t+1],Ti[i]-t+1])
      St[[i]][Ti[i]-t] =  rdisc(n=1, prob=Pt[Ti[i]-t,])
      }
  }
      if (S==1) St[[i]] =  rep(1, Ti[i])
      }
    assign("St", St, envir=.GlobalEnv)
    return(St)
    }

#-------------------------------------------------------------------------------
#--------------------------- Update the emission probability matrix (epm)
#---------------------------

update.epm = function(tying)
{
mij =  list(NULL)
```

```
for (k in 1:K) mij[[k]] =  matrix(0, nrow=L, ncol=S)

for (i in 1:I)
for (s in 1:S)
for (l in 1:L)
mij[[Ci[i]]][l,s] =  mij[[Ci[i]]][l,s]+sum(Yt[[i]]==l & St[[i]]==s)

for (k in 1:K)
for (s in 1:S)
epm[[k]][,s] =  t(rdirichlet(1,alpha=mij[[k]][,s]+tau))

if (tying){
      for (k in 1:(K-1)) mij[[k+1]] = mij[[k]] + mij[[k+1]]
      for (s in 1:S)  epm[[1]][,s] = rdirichlet(1,alpha=mij[[K]][,s]+tau)
      for (k in 1:K)  epm[[k]] = epm[[1]]
 }

assign("epm", epm, envir=.GlobalEnv)
return(epm)
}

#-------------------------------------------------------------------------------
#--------------------------------- Update the initial state probabilities (ispd)
#---------------------------------

update.ispd = function()
{
for (k in 1:K)  ispd[[k]] =  revise.ispd(tpm[[k]])
  assign("ispd", ispd, envir=.GlobalEnv)
return(ispd)
}

#-------------------------------------------------------------------------------
#------------------------------- Update the Membership Probability Matrix (mpm)
#---------------------------

# update.mpm = function()
#  {
#  for (k in 1:K)
#  for (i in 1:I)
#      mpm[i,k] = sum(diag(log(epm[[k]][Yt[[i]],St[[i]]])))
#      mpm = exp(mpm)/apply(exp(mpm), MARGIN=1, FUN=sum)
#   assign("mpm", mpm, envir=.GlobalEnv)
#  return(mpm)
#  }

update.mpm = function()
{
foo = matrix(0,ncol=K, nrow=I)
for (k in 1:K)
for (i in 1:I)
{
 ll.St = sum(log( ispd[[k]][St[[i]]][1]*
   diag(tpm[[k]][St[[i]][1:(Ti[i]-1)],St[[i]][2:Ti[i]]])))
 ll.Yt = sum(diag(log(epm[[k]][Yt[[i]],St[[i]]])))
 ll.Ci = sum(log(omega[k]))
 foo[i,k] = ll.Yt+ll.St+ll.Ci
```

```
}
for (k in 1:K)
  {
    foo2 = foo-matrix(foo[,k], ncol=K, nrow=I)
    mpm[,k] = 1/ apply(exp(foo2), MARGIN=1, FUN=sum)
    }
 mpm[is.nan(mpm)] = 0
  assign("mpm", mpm, envir=.GlobalEnv)
return(mpm)
}

#-------------------------------------------------------------------------------
#--------------------------- Update the Membership variable (Ci)
#---------------------------

update.Ci = function()
{
for (i in 1:I)
    Ci[i] = rdisc(n=1, prob=mpm[i,])
  assign("Ci", Ci, envir=.GlobalEnv)
return(Ci)
}

#-------------------------------------------------------------------------------
#--------------------------- Update the mixture weight (omega)
#---------------------------

update.omega = function()
{
w = numeric(K)
for (k in 1:K)   w[k] = sum(Ci==k)+zeta[k]
omega = rdirichlet(1,alpha=w)
  assign("omega", omega, envir=.GlobalEnv)
return(omega)
}

#-------------------------------------------------------------------------------
#--------------------------- Computing Likelihood
#---------------------------

comp.llik = function()
{
llik = matrix(0, nrow=I, ncol=K)
for (k in 1:K)
    {
    for (i in 1:I)
{
alpha = as.double(ispd[[k]])
lscale = as.double(0)
py = My.ffun(Yt[[i]], epm[[k]])
for (t in 1:Ti[i])
    {
    if (t>1)
alpha = alpha %*% tpm[[k]]
    if (is.vector(py))  alpha = alpha * py else
      alpha = alpha * py[t,]
    sum.alpha = sum(alpha)
```

```
    alpha = alpha/sum.alpha
    lscale = lscale + log(sum.alpha)
    }
llik[i,k] = lscale
}
    }
llik = sum(log(exp(llik)%*%t(omega)))
  assign("llik", llik, envir=.GlobalEnv)
return(llik)
}



#-----------------------------------------------------------------------------
#--------------------------- Send output into the external text file
#---------------------------

send.out = function(tpm.t,epm.t,ispd.t, Ci.t,omega.t, llik.t,St.t)
    {
    write.table(tpm.t,  file = "tpm.csv",   col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    write.table(epm.t,  file = "epm.csv",   col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    write.table(ispd.t, file = "ispd.csv",  col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    write.table(Ci.t,   file = "Ci.csv",    col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    write.table(omega.t,file = "omega.csv", col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    write.table(llik.t, file = "llik.csv",  col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    write.table(St.t,   file = "St.csv",    col.names=FALSE, row.names=FALSE, sep = ",", append=TRUE)
    }


#-----------------------------------------------------------------------
#--------------------------- Main Gibbs sampler program
#---------------------------

MyGibbs = function(Yt,K,S,N,J, burnin, thin, iter, tpm, epm, ispd, St, omega, Ci, mpm, Nr, tying=FALSE,
   tpm0=NULL, epm0=NULL, ispd0=NULL, St0=NULL, omega0=NULL, Ci0=NULL, mpm0=NULL, method='fb')
    {
    invoke.lib()
    assign("Yt",Yt,envir=.GlobalEnv)
    assign("S",S,envir=.GlobalEnv)
    assign("K",K,  envir=.GlobalEnv)
    assign("J",J,  envir=.GlobalEnv)
    assign("N",N,envir=.GlobalEnv)
    assign("Nr", Nr,envir=.GlobalEnv)
    assign("iter",iter,envir=.GlobalEnv)


    Ti  =  unlist(lapply(Yt, FUN=length));  assign("Ti", Ti , envir=.GlobalEnv)
    L   =  max(unlist(Yt));      assign("L", L , envir=.GlobalEnv)
    I  =  length(Yt);    assign("I", I , envir=.GlobalEnv)
    set.vectors(n=(N-burnin)/thin)

    do.sample = function(Yt, iter)
  {
  hyper.par()
  get.init(Ci0=Ci0, epm0=epm0, tpm0=tpm0, St0=St0)
  assign("t0",Sys.time(),envir=.GlobalEnv)
```

```
  j = 0
  for (r in 1:N)
{
if (r%%1==0) cat(r, "\n")
omega = update.omega()
tpm   = update.tpm();
ispd  = update.ispd()
epm   = update.epm(tying)
mpm   = update.mpm()
llik  = comp.llik()
Ci    = update.Ci()
    if (method='standard')   St = update.St.DG()
    if (method='fb')         St = update.St.FB()

if (r > burnin & r%%thin==0)
    {
    j = j+1
    tpm.t[j,]   =  unlist(tpm)
    epm.t[j,]   =  unlist(epm)
    ispd.t[j,]   =  unlist(ispd)
    Ci.t[j,]    =  Ci
    omega.t[j,]  =  omega
    llik.t[j,]  = llik
    St.t[j,] = unlist(St)[round(seq(1,length(unlist(St)),length=10))]
    }
if (j==Nr)
      {
      send.out(tpm.t,epm.t,ispd.t, Ci.t, omega.t, llik.t, St.t)
      j=0
      }

}
  assign("t1",Sys.time(),envir=.GlobalEnv)
  return(list(tpm.t =tpm.t, epm.t=epm.t, ispd.t=ispd.t, Ci.t=Ci.t, omega.t=omega.t,llik.t=llik.t, St.t=St.t))
  }
    out = do.sample(Yt,iter)
    return(out)
    }
```

B.5 Representing Results of MixHMM

```
##------------------------------------------------------------------------------
##------------------------------------------------------------------------------
## Matrix-plot for EPM
##------------------------------------------------------------------------------

trace.epm = function(epm, nc, nr, grid=TRUE)
{
    mat= matrix(1:(nc*nr), ncol=nc,nrow=nr, byrow=FALSE)
    nf = layout(mat =mat,
  widths = rep(0.9*1/nc  ,nc),
  heights= rep((1/nr)      ,nr),
```

```
  respect=TRUE)
    layout.show(nf)
    mar = par('mar')

    for (i in 1:(nr*nc))
      {
      par(mar=c(.2,.3,.2,0))
      Min= min(epm[,i])
      Max= max(epm[,i])
      R  = (Max-Min)
      plot(epm[,i], type='l', ylim=c(0,1),  lwd=1.5,axes=F)
      if (any(i==1:nr))
  mtext(side=2, text= as.character(i), line=2,cex=.7,family='serif', las=2)
      box()
      if(grid) grid(nx=NULL, col=1, lty=2, lwd=.3)
      if (any(i==1:nr))
      axis(2,at=seq(0.2,0.8,.2),labels=seq(0.2,0.8,.2), cex.lab=0.5, cex.axis=0.8, hadj=0, padj=+0.0, las=1, family="seri
      }
    par(mar=mar)
}

##-----------------------------------------------------------------------------
##-----------------------------------------------------------------------------
## Matrix-plot for TPM
##-----------------------------------------------------------------------------

trace.tpm = function(tpm)
{

    nc= sqrt(ncol(tpm))
    mat= matrix(0, ncol=nc+2,nrow=nc+2, byrow=FALSE)
    mat[2:(nc+1),2:(nc+1)] = 1:(nc^2)

    nf = layout(mat =mat,
  widths = c(1/6,rep(1,nc),1/6),
  heights= c(1/6,rep(1,nc),1/6),
  respect=TRUE)
    layout.show(nf)
    mar = par('mar')

    for (i in 1:(nc^2))
      {
      par(mar=c(.3,.3,0,0))
      plot(tpm[,i], type='l', ylim=c(0,1),  axes=F)
      box(lwd=1.5)
      if (any(i==1:nc))
      axis(2,at=seq(0.2,0.8,.2),labels=seq(0.2,0.8,.2), cex.lab=0.5, cex.axis=0.8, hadj=0, padj=+0.0, las=1,
             family="serif", tick=F)
      grid(nx=NULL, col='darkblue', lty=3, lwd=.3)
      }
    par(mar=mar)
    layout(matrix(c(1)))
}

##-----------------------------------------------------------------------------
##-----------------------------------------------------------------------------
## Matrix-plot for TPM to check label switching between mixture components
```

```
##------------------------------------------------------------------------

mix.lswitch.tpm = function(tpm,K)
{
    N= nrow(tpm)
    S= sqrt(ncol(tpm)/K)
    mat = matrix(0, ncol=S+1,nrow=S+2, byrow=FALSE)
    mat[2:(S+1),2:(S+1)] = 1:(S^2)
    mat[S+2,2:(S+1)] = S^2+1
    mat[2:(S+1),1] = S^2+2
    mat[1,2:(S+1)] = S^2+3

    cols=colors()[c(12,34,128,150,435,52,376,151,31,371)]

    nf = layout(mat =mat,
  widths = c(.1/S, rep(1/S,S)),
  heights= c(.8/S, rep(1/S,S), 1/S),
  respect=TRUE)
    layout.show(nf)

    mar = par('mar')

    for (i in 1:S)
    for (j in 1:S)
      {
      par(mar=c(.3,.3,0,0))
      plot(1:N, ylim=c(0,1),  axes=F,col=2, type='n')
      if (i==1)
      mtext(side=2, text= as.character(j), line=2,cex=.7,family='serif', las=2)
      if (j==1)
      mtext(side=3, text= as.character(i), line=1,cex=.7,family='serif', las=1)
      for (k in 1:K)
  {
  lines(tpm[,j+(i-1)*S+(k-1)*(S^2)], col=cols[k])
  box(lwd=1.5)

  if (i==1)
  axis(2,at=seq(0.0,0.8,.2),labels=seq(0.0,0.8,.2), cex.lab=0.5, cex.axis=0.8, hadj=0, padj=+0.0, las=1,
   family="serif", tick=F)


  grid(nx=NULL, col='darkblue', lty=3, lwd=.3)
    }
    }

      par(mar=c(3,.5,2,.1))
      lab = 1:K
      image(matrix(1:K, ncol=1), col=cols[1:K], axes=F)
      text(seq(0,1,length.out=K), rep(0,K), lab, cex=1, col=colors()[c(rep(24,9),1)], font=2)
      box()
      grid(nx=K, ny=0, col=1, lty=1, lwd=1.2)
      mtext('Mixture Component', side=1, cex=1.1, line=0.5,family='serif')

      par(mar=c(0,0,0,0))
      plot(0, type='n', axes=F, ylab='', xlab='')
      mtext(side=2, 'Transition Probabilities', family='serif', cex=1.1, line=3)
```

```
      par(mar=c(0,0,3,0))
      plot(0, type='n', axes=F, ylab='', xlab='')
      mtext(side=3, 'Transition Probabilities', family='serif', cex=1.1, line=1)

      par(mar=mar)
      layout(matrix(c(1)))

}


##------------------------------------------------------------------------
##------------------------------------------------------------------------
## Matrix-plot for EPM to check label switching between hidden status
##------------------------------------------------------------------------

hmm.lswitch.epm = function(epm, nr, nc, K, main=NULL)
{
    N= nrow(epm)
    mat = matrix(0, ncol=K+2,nrow=nr+3, byrow=FALSE)
    mat[2:(nr+1),2:(K+1)] = 1:(nr*K)
    mat[nr+2,2:(K+1)] = nr*K+1
    mat[1:nr+1,1] = nr*K+2

    cols=colors()[c(12,34,128,150,435,52,376,151,31,371)]

    nf = layout(mat =mat,
  widths = c(.3/K, rep(1/K,K), 0.1/K),
  heights= c(.1/nr,rep(1/nr,nr),.9/nr, .05/nr),
  respect=TRUE)
    layout.show(nf)
    mar = par('mar')
    for(k in 1:K)
       {
       par(mar=c(.1,.5,.2,.1))

       for (i in 1:nr)
           {
           plot(1:N, axes=F, ylim=c(0,1), type='n', ylab='', xlab='')
           if (k==1)
           mtext(side=2, text= as.character(i), line=2,cex=.7,family='serif', las=2)
           if (is.null(main)) main2=paste('Component ',as.character(k))
               else main2 = main
           if (i==1)
           mtext(side=3, text=main2, line=.5,cex=1.,family='serif', las=1)
          for (j in 1:nc)
               {
               lines(epm[,i+(j-1)*nr+(k-1)*nr*nc], col=cols[j])
               box(lwd=1.5)
               if (k==1)
                  axis(2,at=seq(0.0,0.8,.2),labels=seq(0.0,0.8,.2), cex.lab=0.5, cex.axis=0.8, hadj=0, padj=+0.0,
                           las=1, family="serif", tick=F)
                if (i==nr)
                 axis(1,at=seq(0,N,1000),labels=seq(0,N,1000), cex.lab=0.5, cex.axis=0.8, hadj=0.5, padj=-2.0,
                     las=1, family="serif", tick=F)
                grid(nx=NULL, col='darkgrey', lty=1, lwd=.5)
               }
       }
```

```
  }

par(mar=c(1,.5,2,.1))
lab = 1:nc
image(matrix(1:nc, ncol=1), col=cols[1:nc], axes=F)
text(seq(0,1,length.out=nc), rep(0,nc), lab, cex=0.9, col=colors()[c(rep(24,9),1)], font=2)
box()
grid(nx=nc, ny=0, col=1, lty=1, lwd=1.2)
mtext('Hidden State', side=1, cex=1.1, line=.5,family='serif')

par(mar=c(0,0,0,0))
plot(0, type='n', axes=F, ylab='', xlab='')
mtext(side=2, 'Emission Probabilities', family='serif', cex=1.1, line=0.5)

    par(mar=mar)
    layout(matrix(c(1)))
}

##-----------------------------------------------------------------------------
##-----------------------------------------------------------------------------
## Matrix-plot for EPM to check label switching between Mixture components
##-----------------------------------------------------------------------------

mix.lswitch.epm = function(epm, nr, nc, K, main=NULL)
{
    N= nrow(epm)
    mat = matrix(0, ncol=nc+2,nrow=nr+3, byrow=FALSE)
    mat[2:(nr+1),2:(nc+1)] = 1:(nr*nc)
    mat[nr+2,2:(nc+1)] = nr*nc+1
    mat[1:nr+1,1] = nr*nc+2

    cols=colors()[c(12,34,128,150,435,52,376,151,31,371)]
    nf = layout(mat =mat,
  widths = c(.1/nc, rep(1/nc,nc), .1/nc),
  heights= c(.2/nr,rep(1/nr,nr),.5/nr, .2/nr),
  respect=TRUE)
    layout.show(nf)
    mar = par('mar')
    for(j in 1:nc)
{
par(mar=c(.1,.5,.2,.1))

for (i in 1:nr)
    {
    plot(1:N, axes=F, ylim=c(0,1), type='n', ylab='', xlab='')
    if (is.null(main))  main2=paste('Hidden State ',as.character(j))  else main2 = main
    if (i==1)  mtext(side=3, text=main2, line=.5,cex=1.,family='serif', las=1)
    for (k in 1:K)
      {
      if (k==1 & j==1) mtext(side=2, text= as.character(i), line=2,cex=.7,family='serif', las=2)
      lines(epm[,i+(k-1)*nr*nc+(j-1)*nr], col=cols[k])
      box(lwd=1.5)
      if (j==1)  axis(2,at=seq(0.0,0.8,.2),labels=seq(0.0,0.8,.2), cex.lab=0.5, cex.axis=0.8, hadj=0, padj=+0.0,
       las=1, family="serif", tick=F)
      if (i==nr) axis(1,at=seq(0,N,1000),labels=seq(0,N,1000), cex.lab=0.5, cex.axis=0.8, hadj=0.5, padj=-2.0,
       las=1, family="serif", tick=F)
        grid(nx=NULL, col='grey', lty=1, lwd=0.5)
```

```
      }
    }
  }

par(mar=c(1,.5,2,.1))
lab = 1:K
image(matrix(1:K, ncol=1), col=cols[1:K], axes=F)
text(seq(0,1,length.out=K), rep(0,K), lab, cex=0.9, font=2)
box()
mtext('Mixture Component', side=1, cex=1.1, line=.5,family='serif')

par(mar=c(0,0,0,0))
plot(0, type='n', axes=F, ylab='', xlab='')
mtext(side=2, 'Emission Probabilities', family='serif', cex=1.1, line=0.5)

    par(mar=mar)
    layout(matrix(c(1)))
}

##-----------------------------------------------------------------------------
##-----------------------------------------------------------------------------
## Matrix-Autocorrelation-plot for TPM
##-----------------------------------------------------------------------------

acf.tpm = function(tpm)
{
   nc= sqrt(ncol(tpm))
    mat= matrix(0, ncol=nc+2,nrow=nc+2, byrow=FALSE)
    mat[2:(nc+1),2:(nc+1)] = 1:(nc^2)

    nf = layout(mat =mat,
  widths = c(1/6,rep(1,nc),1/6),
  heights= c(1/6,rep(1,nc),1/6),
  respect=TRUE)
    layout.show(nf)
    mar = par('mar')

    for (i in 1:(nc^2))
      {
      par(mar=c(.3,.3,0,0))
      acf(tpm[,i], axes=F,lwd=2.0)
      grid(nx=NULL, lwd=0.3, col='black', lty=2)
      box()
      if (any(i==1:nc)) axis(2,at=seq(0.2,0.8,.2),labels=seq(0.2,0.8,.2), cex.lab=0.5, cex.axis=0.8, hadj=0,
       padj=+0.0, las=1, family="serif", tick=F)

      if (any(i==seq(nc,nc^2,nc))) axis(1,at=seq(10,30,10),labels=seq(10,30,10), cex.lab=0.5, cex.axis=1.0,  hadj=0, padj
      }
    par(mar=mar)
    layout(matrix(c(1)))
    }

##-----------------------------------------------------------------------------
##-----------------------------------------------------------------------------
## Trace-plot and acf-plot for Likelihood
##-----------------------------------------------------------------------------
```

```
trace.llik = function(llik)
    {
    Min= min(llik/100)
    Max= max(llik/100)
    R  = (Max-Min)
    plot(llik/100,type='l',  ylab='Log-Likelihood [x100]', xlab='Index', ylim=c(Min-R,Max+R))
    box()
    grid(nx=NULL, lwd=1.0, col='black', lty=2)
    }


##------------------------------------------------------------------------
##------------------------------------------------------------------------
## acf-plot for Likelihood
##------------------------------------------------------------------------

acf.llik = function(llik)
{
acf(llik, main='', ylab='ACF of Log-Likelihood', lwd=3)
box()
grid(nx=NULL, lwd=1.0, col='black', lty=2)
}
```